Parameterized Approximation Schemes for Clustering with General Norm Objectives

Fateme Abbasi
University of Wrocław
Poland
fateme.abbasi@cs.uni.wroc.pl

Sandip Banerjee
University of Wrocław
Poland
sandip.ndp@gmail.com

Jarosław Byrka
University of Wrocław
Poland
jby@cs.uni.wroc.pl

Parinya Chalermsook
Aalto University
Finland
parinya.chalermsook@aalto.fi

Ameet Gadekar
Aalto University
Finland
ameet.gadekar@aalto.fi

Kamyar Khodamoradi
University of Regina
Canada
kamyar.khodamoradi@uregina.ca

Dániel Marx
CISPA Helmholtz Center for Information Security
Germany
marx@cispa.de

Roohani Sharma
Max Planck Institute for Informatics
Germany
rsharma@mpi-inf.mpg.de

Joachim Spoerhase
University of Sheffield
United Kingdom
j.spoerhase@sheffield.ac.uk

***Abstract*—This paper considers the well-studied algorithmic regime of designing a $(1+\epsilon)$-approximation algorithm for a k-clustering problem that runs in time $f(k,\epsilon)\textsf{poly}(n)$ (sometimes called an efficient parameterized approximation scheme or EPAS for short[1]). Notable results of this kind include EPASes in the high-dimensional Euclidean setting for k-center [Badŏiu, Har-Peled, Indyk; STOC'02] as well as k-median, and k-means [Kumar, Sabharwal, Sen; J. ACM 2010].**

Our main contribution is a clean and simple EPAS that settles more than ten clustering problems (across multiple well-studied objectives as well as metric spaces) and unifies well-known EPASes. More specifically, our algorithm gives EPASes in the following settings:

- **Clustering objectives: k-means, k-center, k-median, priority k-center, ℓ-centrum, ordered k-median, socially fair k-median (aka robust k-median), or any other objective that can be formulated as minimizing a monotone (not necessarily symmetric!) norm of the distances of the points from the solution (generalizing the symmetric formulation introduced by Chakrabarty and Swamy [STOC'19]).**
- **Metric spaces: Continuous high-dimensional Euclidean spaces, metrics of bounded doubling dimension, bounded treewidth metrics, and planar metrics.**

Prior to our results, EPASes were only known for vanilla clustering objectives (k-means, k-median, and k-center) and each such algorithm is tailored to work for the specific input metric and clustering objective (e.g., EPASes for k-means and k-center in $\mathbb{R}^d$ are conceptually very different). In contrast, our algorithmic framework is applicable to a wide range of well-studied objective functions in a uniform way, and is (almost) entirely oblivious to any specific metric structures and yet is able to effectively exploit those unknown structures. In particular, our algorithm is *not* based on the (metric- and objective-specific) technique of coresets.

Key to our analysis is a new concept that we call *bounded ϵ-scatter dimension*—an intrinsic complexity measure of a metric space that is a relaxation of the standard notion of *bounded doubling dimension* (often used as a source of algorithmic tractability for geometric problems). Our main technical result shows that two conditions are essentially sufficient for our algorithm to yield an EPAS on the input metric M for any clustering objective:

(i) The objective is described by a monotone norm, and
(ii) the ϵ-scatter dimension of M is upper bounded by a function of ϵ.

***Index Terms*—clustering, parameterized approximation algorithms, scattered dimension, norm clustering.**

I. Introduction

In the class of *k-clustering* problems, we are interested in partitioning n data points into k subsets called clusters, each of which is represented by a center. We aim at minimizing a certain objective based on the distances between the data points and their respective cluster centers. This is among the most fundamental optimization problems that arise routinely in both theory and practice and has received attention from various

Fateme Abbasi, Sandip Banerjee and Jarosław Byrka are supported by Polish National Science Centre (NCN) Grant 2020/39/B/ST6/01641. Parinya Chalermsook is supported by European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 759557). Ameet Gadekar is partially supported by the Academy of Finland (grant number 335715). Dániel Marx is supported by the European Research Council (ERC) consolidator grant No. 725978 SYSTEMATICGRAPH.

research communities, including optimization, data mining, machine learning, and computational geometry. Basic clustering problems such as k-Median, k-Center, and k-Means have been researched for more than half a century and yet remain elusive from many perspectives of computation.

This paper considers a prominent and classic algorithmic regime for k-clustering in which one aims at designing *efficient parameterized approximation schemes* (EPAS)—a $(1+\epsilon)$ approximation algorithm that runs in time $h(k,\epsilon)\mathsf{poly}(n)$ for every $\epsilon > 0$. In a general metric space, obtaining such an approximation scheme is impossible even for basic clustering problems. Past research has therefore focused on designing algorithms that work in *structured* metric spaces (such as planar graphs or Euclidean spaces). In the continuous high-dimensional Euclidean space, EPASes are arguably the "fastest" approximation scheme one can hope for [3], [4], so it is no surprise that research on EPASes for clustering problems has received a lot of attention in the past two decades [5], [6], [7], [8], [9], [10], [11].[2]

This paper is inspired by the following meta-question:

> For a given k-clustering objective and a (structured) metric space, does an EPAS exist?

Systematic understanding about this question has been seriously lacking. While affirmative answers for basic clustering problems such as k-Center, k-Median, and k-Means in the continuous high-dimensional Euclidean space have been shown already two decades ago [13], [1] (recently for structured graph metrics [14], [15], [16], [10], [11]), we do not know of any such result for more complex clustering objectives.

This paper makes substantial progress towards a complete understanding of the above meta-question. In particular, we present a unified EPAS that works for a broad class of clustering objectives (encompassing almost all center-based clustering objectives ever considered by the algorithms community and some new ones that further generalize the existing problems) as well as diverse metric spaces, hence settling many well-studied standalone clustering problems as a by-product.[3] In contrast to the existing approaches (where each algorithm is tailored to specific input metric and clustering objective), our algorithmic framework is (almost) entirely oblivious to any specific metric structures and the objective function, yet is able to effectively exploit those unknown structures.

A. *Efficient Parameterized Approximation Schemes for* Norm k-Clustering

As an input to the (general) k-clustering problem, we are given n data points P, candidate centers F, a metric space $M = (P \cup F, \delta)$, a positive integer k, and an objective function $f\colon \mathbb{R}^P \to \mathbb{R}$. When a set of k "open" centers $X \subseteq F$ is chosen, this solution induces a cost vector $\boldsymbol{\delta}(P,X) = (\delta(p,X))_{p\in P}$ where $\delta(p,X) = \min_{x\in X}\delta(p,x)$ represents the distance from point p to the closest center in X. Our goal is to minimize $f(\boldsymbol{\delta}(P,X))$. We call this problem the k-clustering problem with cost function f. We may think of the function f as "aggregating" the costs incurred by the points. For example, we can formulate basic k-clustering objectives via the functions $f(\boldsymbol{x}) = \sum_{p\in P} x(p)$ (k-Median), $f(\boldsymbol{x}) = \sum_{p\in P} x(p)^2$ (k-Means) and $f(\boldsymbol{x}) = \max_{p\in P} x(p)$ (k-Center).

Most natural and well-studied clustering objectives can be modeled using (a generalization of) the concept of *norm* optimization introduced by Chakrabarty and Swamy [17]. More specifically, we are interested in the setting where the objective f is a norm. A norm is a function $f\colon \mathbb{R}^n \to \mathbb{R}_{\geq 0}$, $n \in \mathbb{N}$ that satisfies (i) for all $\boldsymbol{x} \in \mathbb{R}^n$, $f(\boldsymbol{x}) = 0$ if and only if $\boldsymbol{x} = \boldsymbol{0}$, (ii) $\forall \boldsymbol{x},\boldsymbol{y} \in \mathbb{R}^n\colon f(\boldsymbol{x}+\boldsymbol{y}) \leq f(\boldsymbol{x}) + f(\boldsymbol{y})$, and (iii) $\forall \boldsymbol{x} \in \mathbb{R}^n, \lambda \in \mathbb{R}\colon f(\lambda\boldsymbol{x}) = |\lambda| f(\boldsymbol{x})$. We say that f is *monotone* if $f(\boldsymbol{x}) \leq f(\boldsymbol{y})$ whenever $\boldsymbol{x} \leq \boldsymbol{y}$. By Norm k-Clustering we refer to the k-clustering problem whose objective $f\colon \mathbb{R}^P \to \mathbb{R}_{\geq 0}$ is a monotone norm. While Chakrabarty and Swamy [17] further require that f be symmetric[4], our algorithmic framework applies to all monotone norm cost functions. This family includes the following well-known clustering problems (see Figure 2 for an overview):

- **From k-Means, k-Center, and k-Median to (k,z)-Clustering**: All the basic clustering problems can be captured by the ℓ_z-norm when $z \in \{1,2,\infty\}$. In fact, the (k,z)-clustering problem [18], [19], [20] (for constant positive integer z) uses the objective function $g(\boldsymbol{x}) = \sum_{p\in P}|x(p)|^z$. (This function itself is not a norm, but we can instead consider the ℓ_z-norm $f(\boldsymbol{x}) = g(\boldsymbol{x})^{1/z}$.)
- **Weighted k-Center (or Priority k-Center)**: The weighted version of k-Center [21], [22], [23] generalizes the k-Center so that

[1] Quick remarks: (i) An EPAS is not comparable to polynomial time approximation schemes (PTAS), (ii) before the term EPAS was invented some researchers call this type of approximation schemes a PTAS or simply an approximation scheme (in clustering, it is often assumed that k is small) [1], [2], and (iii) both EPAS and PTAS are implied by the existence of efficient polynomial time approximation schemes (EPTAS).

[2] We remark that PTASes, which are incomparable to EPASes, do not exist for continuous k-means, k-median and k-center [12], [4].

[3] There are variants of clustering problems that enforce constraints on how points can be assigned to open centers (e.g., capacitated and diversity constraints). Our purpose is handling many center-based clustering objectives; handling a broad range of constraints (such as capacities) is beyond the scope of this paper.

[4] We say that f is symmetric if $f(\boldsymbol{x}) = f(\boldsymbol{x'})$ whenever $\boldsymbol{x'}$ can be obtained by reordering coordinates of $\boldsymbol{x}$.

each data point $p \in P$ is associated with a positive weight (or priority) $w(p)$, and the objective is to minimize the (weighted) maximum distance to a center.[5] This problem can be modelled by the "weighted max" norm $f(\boldsymbol{x}) = \max_{p \in P} w(p)x(p)$. One can analogously define the weighted versions of k-MEDIAN and k-MEANS (see, for example, [24]). We remark that the underlying weighted norms are not symmetric.

- **ℓ-CENTRUM**: This problem (sometimes called k-FACILITY ℓ-CENTRUM) aims to minimize the sum of the connection costs among the ℓ "most expensive" points (that is, those that are furthest away from the open centers). The problem generalizes both k-CENTER ($\ell = 1$) and k-MEDIAN ($\ell = |P|$) problem [25]. (See the books [26], [27] for more details on ℓ-CENTRUM and the more general ORDERED k-MEDIAN discussed below.) This problem can be modelled by the top-ℓ norm $f(\boldsymbol{x}) = \sum_{j=1}^{\ell} x^{\downarrow}(j)$ where $\boldsymbol{x}^{\downarrow}$ denotes the reordering of vector $\boldsymbol{x}$ so that the entries appear non-increasingly. The top-ℓ norm is symmetric.
- **ORDERED k-MEDIAN**: This problem further generalizes ℓ-CENTRUM, allowing flexible penalties to be applied to data points that incur the highest connection costs. More formally, the objective is the *ordered weighted norm* $f(\boldsymbol{x}) = \boldsymbol{v}^{\intercal}\boldsymbol{x}^{\downarrow}$ where $\boldsymbol{v} \in \mathbb{R}^{n}_{\geq 0}$ is a non-increasing cost vector, that is, $v(1) \geq v(2) \geq \ldots \geq v(n)$. ℓ-CENTRUM corresponds to $\boldsymbol{v} = (1, \ldots, 1, 0, \ldots 0)$ where the first ℓ-entries of $\boldsymbol{v}$ are ones. This problem has already received attention for a few decades [28], [17], [29]. We remark that the f here is a monotone and symmetric norm.
- **SOCIALLY FAIR k-MEDIAN (or ROBUST k-MEDIAN)**: In SOCIALLY FAIR k-MEDIAN, along with the point set P, we are given m different (not necessarily disjoint) subgroups such that $P = \bigcup_{i \in [m]} P_i$. Our goal is to find a set X of centers that incurs fair costs to the groups by minimizing the maximum cost over all the groups. In other words,

$$\min_{\substack{X \subseteq F \\ |X| = k}} \max_{i \in [m]} \sum_{p \in P_i} \delta(p, X)\,.$$

Due to distinct applications in at least two domains, this variant of clustering has recently been studied extensively: (i) in algorithmic fairness [30], [31], [32], [33] and (ii) in the robust optimization context, this problem is known as ROBUST k-MEDIAN, which intends to capture the applications when we are uncertain about the actual data scenarios (corresponding to the groups P_i) that may come up [34], [35], [36]. The resulting norm is generally asymmetric.
- **(z, q)-FAIR CLUSTERING:** Our problem also models a clustering problem called (z, q)-FAIR CLUSTERING[6] introduced by Chlamtáč et al. [37], which generalizes SOCIALLY FAIR k-MEDIAN.
In particular, one can view the cost function f of SOCIALLY FAIR k-MEDIAN as a "two-level" aggregate cost: First, cost $\sum_{p \in P_i} \delta(p, X)$ incurred by group P_i, $i \in [m]$ can be viewed as weighted ℓ_1-norm $\boldsymbol{w}_i^{\intercal}\boldsymbol{x}$ where $\boldsymbol{w}_i = \mathbf{1}_{P_i} \in \{0,1\}^P$ denotes the characteristic vector of P_i. Second, these group costs are further aggregated through ℓ_∞, that is, $f(\boldsymbol{x}) = \max(\boldsymbol{w}_1^{\intercal}\boldsymbol{x}, \boldsymbol{w}_2^{\intercal}\boldsymbol{x}, \ldots, \boldsymbol{w}_m^{\intercal}\boldsymbol{x})$.
(z, q)-FAIR CLUSTERING allows arbitrary uses of ℓ_z and ℓ_q norms to aggregate the costs in two levels. The cost function is defined as $f(\boldsymbol{x}) = g(\boldsymbol{h}(\boldsymbol{x}))$ where g is any ℓ_q-norm function and $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), h_2(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x}))$ where $h_i(\boldsymbol{x})$ is a weighted ℓ_z-norm, that is, $h_i(\boldsymbol{x}) = \left(\sum_{p \in P} w_i(p)x(p)^z\right)^{1/z}$ for arbitrary weight vectors $\boldsymbol{w}_i \in \mathbb{R}^{P}_{\geq 0}$, $i \in [m]$. It is easy to check that $f(\boldsymbol{x}) = g(\boldsymbol{h}(\boldsymbol{x}))$ is a monotone norm whenever g and $\{h_i\}$ are. The objective is generally an asymmetric norm.
- **Beyond the Known Problems**: Our (asymmetric) norm formulation allows us to model more complex clustering objectives that might be useful in some application settings and, to our knowledge, have not yet been considered in the algorithms community. One such objective is PRIORITY ORDERED k-MEDIAN: We have the cost function $f(\boldsymbol{x}) = \boldsymbol{v}^{\intercal}\boldsymbol{x_w}^{\downarrow}$ where the weight vector $\boldsymbol{v} \in \mathbb{R}^{n}_{\geq 0}$, and priority vector $\boldsymbol{w} \in \mathbb{R}^{P}_{\geq 0}$ are given as input, and where $\boldsymbol{x_w} = (w(p)x(p))_{p \in P}$. This objective generalizes both PRIORITY k-CENTER and ORDERED k-MEDIAN. Another natural objective is the (multi-level) CASCADED NORM CLUSTERING, which generalizes (z, q)-FAIR CLUSTERING to allow multiple levels of cost aggregation. The cost function f for this problem is described by a directed acyclic graph (DAG) D with one sink node and $|P|$ source nodes (each source corresponds to a point in P). Each non-source node v is associated with a norm ℓ_q for some q, and each edge (u, v) has weight $w_{u,v}$. Given such a DAG D, the value of $f(\boldsymbol{x})$ can be evaluated by computing the evaluations at nodes in $V(D)$ in (topological) order from sources to sink: (i) The evaluation at source

[5] For convenience of presentation, the terminologies we use are somewhat different from the literature.

[6] Chlamtáč et al. [37] call the problem (p, q)-FAIR CLUSTERING. For the sake of consistency with the notation in the rest of the paper, we changed the naming slightly.

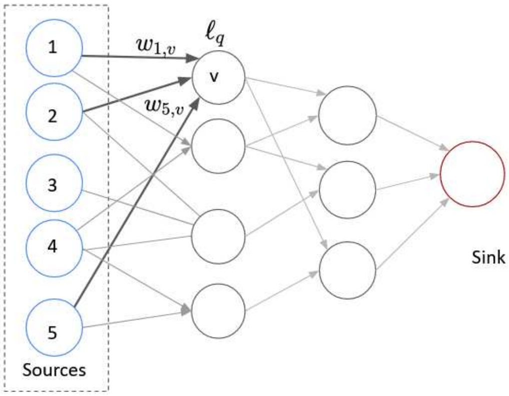

Fig. 1: The DAG here describes evaluation of function f. Node v is labeled with the ℓ_q norm, so the evaluation at node v is $\eta(v) = (w_{1,v}x_1^q + w_{2,v}x_2^q + w_{5,v}x_5^q)^{1/q}$.

$p \in P$ is $\eta(p) = x(p)$, (ii) For any non-source node $v \in V(D)$ labelled with the norm ℓ_q, we evaluate $\eta(v) = \left(\sum_{u \in N^-(v)} w_{u,v}\eta(u)^q\right)^{1/q}$, and (iii) the value of $f(\boldsymbol{x})$ is the evaluation of the sink. See Figure 1 for illustration. (z,q)-FAIR CLUSTERING is a special case when D has 3 layers with the middle layer using the same norm. Of course, also other basic monotone norms such as top-ℓ or ordered weighted norms could be composed to more complex norms analogously.

We remark that *asymmetric* norms can potentially make the problem substantially harder. For example, a poly-time $\mathcal{O}(1)$-approximation algorithm exists for symmetric norms [17] but the asymmetric norm makes it $\Omega(\log n/\log\log n)$-hard to approximate even for the special case of ROBUST k-MEDIAN on the line metrics [35].

Our main results are encapsulated in the following theorem.

Theorem I.1. *Let f be an efficiently (approximately) computable monotone norm cost function. Then the k-clustering problem with cost function f admits an EPAS for the following input metrics: (i) metrics of bounded doubling dimension, (ii) continuous Euclidean spaces of any dimension, (iii) bounded treewidth metrics, and (iv) planar metrics.*

By *continuous* Euclidean space, we refer to the setting where any point of the space can be chosen as a center. This is in contrast to a *discrete* Euclidean space, where we restrict the centers to be selected from a specific finite subset of the points. Observe that for a fixed d, discrete Euclidean problems in $\mathbb{R}^d$ have bounded doubling dimension, hence covered by our framework. Furthermore, it is not a shortcoming of our result that it does not cover discrete Euclidean spaces of high dimensions: in this setting, k-CENTER is W[1]-hard to approximate within a factor of $\sqrt{3/2} - o(1)$ [38]. In the setting of graph metrics (such as bounded treewidth or planar metrics), the metric is induced by a graph of the given graph class, the set P of data points and the set F of centers are arbitrary node subsets, and the distance function δ is the shortest-path distance in the graph.

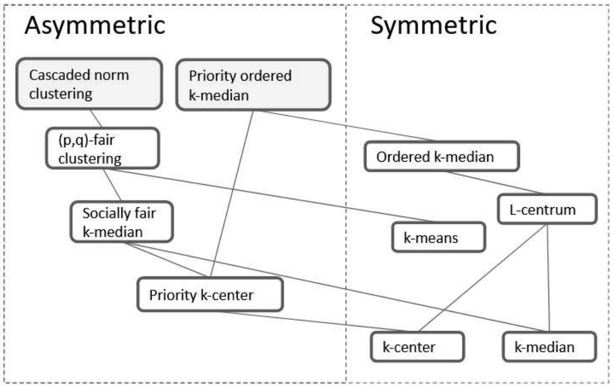

Fig. 2: Selected clustering objectives that can be formulated as monotone norm minimization. The line illustrates generalization (bottom is a special case of top).

Our result in particular implies the following.

Corollary I.2. *In all aforementioned metric spaces,*

1) *There exists a $2^{h(1/\epsilon)\cdot k\cdot \textit{polylog}(k)} \cdot \textit{poly}(n)$ time EPAS for* ORDERED k-MEDIAN *on n points.*
2) *There exists a $2^{h(1/\epsilon)\cdot k\cdot \textit{polylog}(k)} \cdot \textit{poly}(n,m)$ time EPAS for (z,q)-*FAIR CLUSTERING *on n points and m groups.*

Prior to our results, the existences of EPASes for all these problems were open (except for k-MEANS, k-CENTER, and k-MEDIAN). Beyond these known problems, we also obtain EPASes for the new, generalized problems introduced above and depicted in Figure 2. Rather surprisingly, in contrast to the poly-time approximation regime, the complexities of symmetric and asymmetric norm clustering problems “collapse” in the parameterized approximation regime.

B. Our Conceptual and Technical Contributions

Our main contributions have two parts: (i) a new concept of metric dimension and (ii) our main technical result showing EPASes for all the aforementioned clustering problems.

Unifying Metric Spaces via Scatter Dimension: Our key conceptual contribution is a new notion of bounded metric space dimension that relaxes the standard requirement of bounded doubling dimension so that the metric spaces mentioned in Theorem I.1 all "live" in a finite dimension. We first explain why existing notions of dimensions are not suitable for such purpose.

There are multiple dimensionality notions that appear in the literature of metric spaces. Most familiar in the algorithmic community is perhaps the *doubling dimension* (a.k.a. Assouad dimension). Roughly, the doubling dimension of metric (M, δ) is $\mathcal{O}(d)$ iff at most $(1/\epsilon)^{\mathcal{O}(d)}$ balls of radius $\epsilon/2$ can be packed into a unit ball (this is called an ϵ-packing). Such property can often be computationally leveraged, leading to efficient algorithms for many geometric optimization problems (often with the running time depending exponentially on the dimension). However, the doubling dimension (as well as any other popular notions of dimensions [39]) would not be suitable for us due to the following reasons: (i) The doubling dimension can be as large as $\Omega(n)$ in high-dimensional Euclidean space, and (ii) they do not very well "exploit" structured graph metrics, i.e., even stars have unbounded dimension.[7] In sum, any algorithms that exploit existing notions of dimensions are unlikely to lead to our desired results.

We introduce the notion of *ϵ-scatter dimension*. Given metric $M = (P, F, \delta)$, the sequence $(x_1, p_1), \ldots, (x_\ell, p_\ell) \in F \times P$ is said to be an ϵ-scattering if, whenever (x, p) appears before (x', p') in the sequence, then $\delta(x, p)$ and $\delta(x', p')$ are larger than $1 + \epsilon$ each, while $\delta(x', p) \leq 1$. The ϵ-scatter dimension of M is then defined as the length of the longest scatter, minus one.

There are two natural interpretations. The first interpretation is as a game between two players: The *center player* who tries to claim she can cover all the points with a unit ball and the *point player* who present a counterexample. In the first round, the center player picks a center $x_1 \in F$ and the point player *refutes* the claim by presenting a point $p_1 \in P$ which is at least a factor $1 + \epsilon$ away from the (closed) unit ball around x_1, that is, $p_1 \notin \mathsf{ball}(x_1, 1+\epsilon)$. The game continues this way: In the i-th round, the center player presents x_i such that $\{p_1, \ldots, p_{i-1}\} \subseteq \mathsf{ball}(x_i, 1)$, and the point player gives $p_i \notin \mathsf{ball}(x_i, 1 + \epsilon)$. Both players are interested in prolonging the game as much as possible. The ϵ-scatter dimension is the length of the longest possible game. In the second interpretation, one can view such sequence as a pair of ϵ-packings that are required to be sufficiently distanced: It is easy to verify (simply using triangle inequalities) that $P^* = \{p_1, p_2, \ldots, p_{\ell-1}\}$ and $F^* = \{x_2, \ldots, x_\ell\}$ are ϵ-packings of the unit (closed) balls around x_ℓ and p_1, respectively. This view immediately implies that ϵ-scatter dimension is bounded in a bounded doubling metric.

Theorem I.3. *For $\epsilon \in (0, 1)$, any metric of doubling dimension d has ϵ-scatter dimension $(1/\epsilon)^{\mathcal{O}(d)}$.*

We proceed to study the ϵ-scatter dimension of graph metrics where we the set P of data points and the set F of centers are arbitrary node subsets in a graph of some fixed graph class and the distances between them are the shortest path distances.

Theorem I.4. *For $\epsilon \in (0, 1)$, the ϵ-scatter dimension is $\exp\left((1/\epsilon)^{\mathcal{O}(\mathtt{tw})}\right)$ for treewidth-$\mathtt{tw}$ graphs.*

This proof is based on a (delicate) combinatorial argument that, given graph G, parameter t and an ϵ-scattering sequence of length at least doubly exponential in t, produces a "certificate" to the fact that the treewidth of G is greater than t. The proof can be found in Section VI-B.

Next, we present a tool that allows "bootstrapping" of graph classes having bounded ϵ-scatter dimension. This is done via a simple connection between ϵ-scatter dimension and low-treewidth embedding (an active area of metric space embedding) [40], [16], [41]. This connection would allow us to reduce the question of bounding ϵ-scatter dimension in a certain graph class to that in bounded treewidth graphs (thereby invoking our Theorem I.4.)

Theorem I.5 (informal, formal statement in Section VI-C)**.** *The ϵ-scatter dimension is bounded for any graph class $\mathcal{G}$ that admits an η-additive distortion embedding (error $\pm\eta\Delta$ where Δ is the diameter of the graph) into a graph whose treewidth only depends on η.*

Such a connection, combined with the embedding result of [16], implies the following. (Again we allow the sets of points and centers to be arbitrary node subsets.)

Theorem I.6. *For $\epsilon \in (0, 1)$, the ϵ-scatter dimension is $\exp\left(\exp(\mathsf{poly}(1/\epsilon))\right)$ for planar graphs.*

Moreover, further progresses in the area of low-treewidth embedding would lead to even wider classes of graphs that have bounded ϵ-scatter dimension, e.g., it seems plausible that minor-free graphs admit such an embedding [40].

Unfortunately, the bounded dimensionality does not hold in the high-dimensional (continuous) Euclidean metric.[8] To handle the high-dimensional continuous Eu-

[7] In an n-node star rooted at r, a unit ball $\mathsf{ball}(r, 1)$ includes the whole graph. There exists an ϵ-packing of size $(n-1)$ by choosing the non-root nodes.

[8] To see this, consider the sequence $(x_1, p_1) \ldots, (x_{d-1}, p_{d-1})$ where, for each $i \in [d-1]$, the point $x_i \in \mathbb{R}^d$ has i-th coordinate $1/\sqrt{2}$ and all other coordinates are zero. Define points $p_i = -x_i$ for all $i \in [d-1]$. It is easy to check that this sequence is a $(\sqrt{2}-1)$-scattering. This example implies that the ϵ-scatter dimension of continuous Euclidean metrics $\mathbb{R}^d$ can be at least $d-1$ (unbounded in ϵ). In fact, the ϵ-scatter dimension is as high as $(1/\epsilon)^{\Omega(d)}$.

clidean setting, we present a stronger version of ϵ-scatter dimension, that we call *algorithmic ϵ-scatter dimension*. The setting of the game is the same except that the center player would *optimize* to end the game early, while the point player would be interested in prolonging the game indefinitely. This means, they play against each other. A *centering strategy* is a function $\sigma\colon 2^P \to F$ that specifies how the center $x_i = \sigma(\{p_1, \ldots, p_{i-1}\})$ would be chosen by the center player, given the points $p_1, \ldots, p_{i-1}$ played in the preceding rounds. The (σ, ϵ)-scatter dimension is the maximum number of rounds when the center player always plays strategy σ, and the algorithmic ϵ-scatter dimension is the minimum (σ, ϵ)-scatter dimension over all strategies σ. We remark that our actual definition is more involved, as it considers a weighted version of the game.

Theorem I.7 (Bounding Algorithmic Scatter Dimension)**.** *The continuous Euclidean space (P, F, δ), that is, $P \subsetneq \mathbb{R}^d$ finite, and $F = \mathbb{R}^d$, has algorithmic ϵ-scatter dimension $\mathcal{O}(1/\epsilon^4 \log 1/\epsilon)$.*

EPAS for General Norm Clustering: Bypassing Coresets: Now we are ready to explain our main technical result that would allow us to obtain EPAS for all metrics having bounded ϵ-scatter dimension.

A generic tool whose existence immediately implies an EPAS is an ϵ-coreset —a "compression" of an input instance (P, F, δ) into a much smaller instance so that the cost of any solution is preserved within a factor of $(1 \pm \epsilon)$. The existence of an ϵ-coreset of size depending only on ϵ and k would immediately imply an EPAS (but not vice versa): First, use the ϵ-coreset to compress the instance (P, F, δ) to (P', F', δ') where $|P'| \leq \gamma(\epsilon, k)$. Then enumerate all possible partitionings of P' into k sets $P'_1, \ldots, P'_k$ (there are at most $k^{\gamma(\epsilon,k)}$ such partitions). For each set $i \in [k]$, compute the optimal center for P'_i. We choose the partition that gives the lowest total cost.

This generic method, unfortunately, faces a serious information-theoretic limitation, that is, even for k-CENTER, ϵ-coresets of desirable sizes do not exist in high-dimensional Euclidean spaces [29]. Such lower bounds imply that one cannot hope to prove our (unified) results via the coreset route: While coresets are known for (k, z)-CLUSTERING for constant z [19]—allowing to handle k-MEANS and k-MEDIAN in a uniform fashion—it is impossible to extend this approach to k-CENTER. For more complex clustering objectives, such EPASes were in fact not known even for low dimension. For example, the coreset of Braverman et al. [29] for ORDERED k-MEDIAN in $\mathbb{R}^d$ has size $\mathcal{O}_{\epsilon,d}(k^2 \log^2 n)$ and therefore does not give an EPAS even in low dimension.

Badoiu, Har-Peled, and Indyk [13] presented an EPAS for k-CENTER in high-dimensional Euclidean spaces (bypassing coresets in the above sense). Therefore, an obvious open question is whether their techniques can be extended to give an EPAS for any other clustering objective. Unfortunately, this is not even known for simple objectives such as PRIORITY k-CENTER. In fact, even the known EPASes for k-MEANS [1] and k-CENTER [13] are conceptually very different; to our knowledge, no approximation schemes handle k-MEANS and k-CENTER in a modular way.

Our main technical result is presented in the following theorem. We remark that our techniques do not rely on any coreset constructions (thus bypassing the coreset lower bounds for k-CENTER).

Theorem I.8. *Let $\mathcal{M}$ be a class of metric spaces that is closed under scaling distances by a positive constant. There is a randomized algorithm that computes for any* NORM k-CLUSTERING *instance $\mathcal{I} = (M, f, k)$ with metric $M = (P, F, \delta) \in \mathcal{M}$, and any $\epsilon \in (0, 1)$, with high probability a $(1 + \epsilon)$-approximate solution if the following two conditions are met.*

(i) There is an efficient algorithm evaluating for any distance vector $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ the objective $f(\boldsymbol{x})$ in time $T(f)$.

(ii) There exists a function $\lambda\colon \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $\epsilon > 0$, the algorithmic ϵ-scatter dimension of $\mathcal{M}$ is at most $\lambda(\epsilon)$.

The running time of the algorithm is $\exp\left(\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right) \cdot \mathsf{poly}(|M|) \cdot T(f)$.

Note that the complexity of computing f appears only as a linear factor in the running time. For instance, for SOCIALLY FAIR k-MEDIAN, the number m of groups affect only the computational cost of f, and therefore the running time is polynomial in m. We remark that our results extend to the setting of an approximate evaluation oracle where f can be computed to within a factor $1 \pm \epsilon$ in time $T(f)\mathsf{poly}(1/\epsilon)$ where $T(f)$ depends only on f but not on ϵ. For the sake of easier presentation we assume in this conference proceedings version that we have an exact evaluation oracle for f.

Our algorithm is clean, simple, and entirely oblivious to both the objective and the structure of the input metric.

The dependency on k in the exponent of our running time is singly exponential ($\exp(\widetilde{\mathcal{O}}_\epsilon(k))$). In terms of k, we therefore match the running time of the fastest known EPAS for the highly restrictive special case of high-dimensional k-MEANS [1]. Moreover, the dependency on ϵ in the exponent could be improved by proving better bounds on the ϵ-scatter dimension of a metric space of interest, e.g., $\lambda(\epsilon) = \mathsf{poly}(1/\epsilon)$ implies the EPAS running time $\exp(\widetilde{\mathcal{O}}(k) \cdot \mathsf{poly}(1/\epsilon))$.

II. Overview of Techniques

In this section, we give an informal overview of the technical ideas appearing in the paper. The main result will be built step by step: we believe that it is already interesting to understand our main result specialized to WEIGHTED k-CENTER and WEIGHTED k-MEDIAN. Our starting point is the EPAS of Badŏiu et al. [13] for unweighted k-CENTER that works on high-dimensional Euclidean spaces. We redesign and change this algorithm in order to be able to present it with a clean division into two parts: a simple branching algorithm and a bound on the abstract concept of (algorithmic) ϵ-scatter dimension. This way, we obtain a sharp separation between the branching algorithm, which is specific to the objective and the bound on ϵ-scatter dimension, which is specific to the metric. This can be contrasted with techniques based on coresets, which are inherently specific both to a single objective and to a single metric. The main message of the paper is that, with the right combination of additional ideas, this framework can be significantly generalized both in terms of objectives and metric spaces.

This section presents the main algorithmic ideas in three steps.

1) The algorithm for unweighted k-CENTER can be generalized to WEIGHTED k-CENTER in a not completely obvious way.
2) Building on the algorithm for WEIGHTED k-CENTER, we can solve WEIGHTED k-MEDIAN with a preprocessing and a random selection step.
3) The WEIGHTED k-MEDIAN algorithm can be generalized to arbitrary monotone norms by considering infinitely many WEIGHTED k-MEDIAN instances defined by the subgradients.

While some of the challenges on the way may appear to have other approaches promising at first glance, we want to emphasize that it is nontrivial to find the combination of ideas that can be integrated together to obtain our main result. In particular, for WEIGHTED k-MEDIAN the initial upper bounds have to be defined carefully in a way that allows, at the same time, an efficient random selection step and generalization to arbitrary monotone norms.

a) WEIGHTED k-CENTER with Bounded Number of Different Weights: Our starting point is a simple branching algorithm that is inspired by the EPAS of Badŏiu et al. [13] for unweighted k-CENTER. Instead of branching, it will be more convenient for us to present it as a randomized algorithm. Furthermore, we consider the more general setting of WEIGHTED k-CENTER: the objective is to find a set O of k centers that minimizes $\max_p w(p)\delta(p, O)$. Let us first present the algorithm with the simplifying assumption that w is a weight function on the points whose range contains only at most τ different values. The unweighted problem corresponds to $w(p) = 1$ for every $p \in P$ and hence $\tau = 1$. It will be convenient to assume that we (approximately) know the value of OPT.

We start with k arbitrarily chosen candidates $X = \{x_1, \ldots, x_k\}$ for the k centers. We additionally introduce k sets of *requests* $Q_1, \ldots, Q_k$, where each request is of the form (p, r) with a point $p \in P$ and radius $r > 0$. For every $\kappa \in [k]$, we impose the *cluster constraint* requiring that, for every $(p, r) \in Q_\kappa$, center x_κ should be at distance at most r from p. Initially, we set $Q_\kappa = \emptyset$ for every κ, which means that these conditions are trivially satisfied. If we have $\max_p w(p)\delta(p, X) \le (1+\epsilon)\mathsf{OPT}$, then we can stop, as we have a $(1+\epsilon)$-approximate solution at our hands. Otherwise, we have a point p with $\delta(p, X) > (1+\epsilon)\mathsf{OPT}/w(p)$, while it is at distance at most $\mathsf{OPT}/w(p)$ from some center of a hypothetical optimum solution O. Thus the algorithm selects a $\kappa \in [k]$ uniformly at random, hoping it to be the index of the center that is at distance at most $\mathsf{OPT}/w(p)$ from p in the optimum solution O. Then we introduce the request $(p, \mathsf{OPT}/w(p))$ into the set Q_κ and select x_κ to be a center that satisfies the cluster constraint defined by all the requests in the updated Q_κ. Observe that if every random choice was compatible with the hypothetical optimum solution O, then the algorithm is always able to find such a center, as the requests in Q_κ are always satisfied by the κ-th center of the optimum solution O.

We claim that if the ϵ-scatter dimension of the metric is bounded, then this algorithm stops after a bounded number of steps, either by finding an approximate solution or by failing to find a center satisfying the cluster constraints of some Q_κ. Let $x_\kappa^{(1)}, \ldots, x_\kappa^{(\ell)}$ be the different candidates for the κ-th center throughout this branch. Let $(p_\kappa^{(1)}, r_\kappa^{(1)}), \ldots, (p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ be the requests introduced to Q_κ: that is, for $1 \le j \le \ell$, the center $x_\kappa^{(j)}$ was chosen to be at distance at most $r_\kappa^{(i)}$ from every $p_\kappa^{(i)}$ for $1 \le i < j$, but later was found to be at distance at least $(1+\epsilon)r_\kappa^{(j)}$ from $p_\kappa^{(j)}$. As there are at most τ different weights in the input, at least $\ell' = \ell/\tau$ of these requests have the same radius. That is, there is a subsequence $(x_\kappa^{(s_1)}, p_\kappa^{(s_1)}, r_\kappa^{(s_1)}), \ldots, (x_\kappa^{(s_{\ell'})}, p_\kappa^{(s_{\ell'})}, r_\kappa^{(s_{\ell'})})$ where every $r_\kappa^{(s_j)}$ for $j \in [\ell']$ is the same value $r \ge 0$. This means that we have a subsequence $(\bar{x}_1, \bar{p}_1), \ldots, (\bar{x}_{\ell'}, \bar{p}_{\ell'})$ with the property that $\delta(\bar{x}_i, \bar{p}_i) > (1+\epsilon)r$, but $\delta(\bar{x}_i, \bar{p}_j) \le r$ for every $i < j$. By scaling down every distance by a factor of r, this is precisely an ϵ-scattering of length ℓ'. If we consider a class of metrics closed under scaling where the ϵ-scatter dimension is $\lambda(\epsilon)$, then this sequence cannot have length longer than $\lambda(\epsilon)$, implying that $\ell \le \tau \cdot \lambda(\epsilon)$. We can conclude that the algorithm can introduce at most

$\tau \cdot \lambda(\epsilon)$ requests into each Q_κ, hence the algorithm cannot perform more than $k \cdot \tau \cdot \lambda(\epsilon)$ iterations.

If every step of the algorithm randomly chooses an index $\kappa \in [k]$ that is consistent with the optimum solution O, then the only way it can stop is by finding an approximate solution. Therefore, the algorithm is successful with probability at least $q = k^{-k \cdot \tau \cdot \lambda(\epsilon)}$. The success probability can be boosted to be a constant arbitrarily close to 1 by the standard technique of repeating the algorithm $\mathcal{O}(1/q)$ times, leading to a running time of $k^{k \cdot \tau \cdot \lambda(\epsilon)} \cdot \mathsf{poly}(n)$.

b) WEIGHTED k-CENTER *with Arbitrary Weights:* We show now how the algorithm can be extended to work in the weighted setting with arbitrary weights. Let us observe first that if there is no bound on the number τ of different weights, then we cannot bound the number of requests to a given Q_κ, even in very simple metric spaces such as $\mathbb{R}^1$. Suppose for example that the requests arriving to Q_κ are $(p^{(i)}, (1+2\epsilon)^{-i})$ for $i = 1, 2, \ldots$, where every $p^{(i)}$ is at the origin (or maybe within a very small radius of the origin). Then a center $x^{(i)}$ at $(1+2\epsilon)^{1-i}$ satisfies the first $i-1$ requests, but violates the constraint of the i-th by more than a $(1+\epsilon)$-factor. This sequence can be arbitrarily long, and the existence of such a sequence shows that we cannot bound the number of requests arriving to Q_κ if we don't have a bound on the number of different weights. Nevertheless, we show that the number of requests can be bounded if we start the algorithm by carefully seeding the initial requests. Let us remark that we know other simple modifications that achieve such a bound, but the technique described below turns out to be the one that can be extended further for WEIGHTED k-MEDIAN and general norms.

The main idea is to bootstrap our algorithm with a constant-factor approximation. A simple greedy 3-approximation can be obtained following the ideas of Plesník [23]. Let us consider all the balls $\mathsf{ball}(p, \mathsf{OPT}/w(p))$ for every $p \in P$. Let us consider these balls in a nondecreasing order of radius, and mark each ball that does not intersect any of the balls marked earlier; let $\mathsf{ball}(p_\kappa, \mathsf{OPT}/w(p_\kappa))$, $1 \le \kappa \le k'$ be the marked balls. We should have $k' \le k$: otherwise, we have more than k pairwise disjoint balls and each of them has to contain a center of the solution, contradicting the assumption that value OPT can be achieved with k centers. For $1 \le \kappa \le k'$, let x_i be any center in $\mathsf{ball}(p_\kappa, \mathsf{OPT}/w(p_\kappa))$ and let $Q_\kappa = \{(p_\kappa, \mathsf{OPT}/w(p_\kappa)\}$. For $k' < \kappa \le k$, we choose x_i arbitrarily and let $Q_\kappa = \emptyset$. Let us observe that with this definition of the Q_κ's, we have $\delta(p, X) \le 3\mathsf{OPT}/w(p)$ during *every iteration* of our algorithm. Indeed, if the ball of p was marked, then X always contains a center in $\mathsf{ball}(p_\kappa, \mathsf{OPT}/w(p_\kappa))$; if the ball of p was unmarked, then it intersects a marked ball with not larger radius that contains a center of X.

The main claim is that the ratio between the radii of two requests appearing in Q_κ can be bounded by $\mathcal{O}(1/\epsilon)$. Suppose that (p, r) and (p', r') are two requests in Q_κ (introduced in any order) and we have $r' < \epsilon r/4$. A center of the optimum solution satisfies both request, hence we have $\delta(p, p') \le r + r'$. As shown above, at every step of the algorithm there is a center in X at distance at most $3r'$ from p'; let y be such a center at the step when request (p, r) was introduced. Then we have

$$\delta(p, y) \le \delta(p, p') + \delta(p', y) \le r + r' + 3r' \le (1+\epsilon)r,$$

contradicting the need for the first request.

We can use the standard assumption that every weight is of the form $(1+\epsilon)^i$ for some integer i: by rounding down every weight to the largest number of this form, we change the objective only by a factor of $1+\epsilon$. If every weight is of the form $(1+\epsilon)^i$, then the $\mathcal{O}(1/\epsilon)$ bound proved above implies that the requests introduced into Q_κ for some fixed $\kappa \in [k]$ have $\mathcal{O}(1/\epsilon \cdot \log 1/\epsilon)$ different radii. Therefore, we can bound the total number of requests (and hence the number of iterations) by $\mathcal{O}(\lambda(\epsilon) \cdot k/\epsilon \cdot \log 1/\epsilon)$. This leads to a $k^{\mathcal{O}(\lambda(\epsilon) \cdot k/\epsilon \cdot \log 1/\epsilon)} \cdot \mathsf{poly}(n)$ time randomized algorithm with constant success probability.

c) From WEIGHTED k-CENTER *to* WEIGHTED k-MEDIAN*:* Towards our goal of understanding general norms, let us consider now the WEIGHTED k-MEDIAN problem, where the objective is to find a set O of k centers that minimize $\sum_p w(p)\delta(p, O)$. We will try to solve this problem by interpreting it as a WEIGHTED k-CENTER problem on a weighted point set that we dynamically discover during the course of the algorithm.

We would like to turn the *linear constraint* $\sum_p w(p)\delta(p, X) \le \mathsf{OPT}$ of WEIGHTED k-MEDIAN into a *distance constraint:* some point p should be at distance at most r to the solution. Let X be the current solution and suppose that $\sum_p w(p)\delta(p, X) > (1+\epsilon)\mathsf{OPT}$. The intuition is that $\sum_p w(p)\delta(p, X) > (1+\epsilon)\sum_p w(p)\delta(p, O)$ for an optimum solution O implies that a nontrivial fraction of the points should satisfy $\delta(p, X) > (1+\epsilon/3)\delta(p, O)$, that is, their distances to the solution has to be improved by more than a factor of $1+\epsilon/3$. More precisely, an easy averaging argument shows if we select a point p with probability proportional to $w(p)\delta(p, X)$, then p satisfies $\delta(p, X) > (1+\epsilon/3)\delta(p, O)$ with probability $\Omega(\epsilon)$. We call such a point p an $\epsilon/3$-*witness,* certifying that the current solution has to be improved.

Assuming that the sampled point p is indeed a $\epsilon/3$-witness, we proceed as in the case of WEIGHTED k-CENTER. We randomly choose an index κ and introduce the request $(p, \delta(p, X)/(1+\epsilon/3))$ into Q_κ, to update the cluster constraint by requiring that x_κ should be closer to p than in the current solution. If there is a center satisfying all the requests in Q_κ, then we update x_κ. These steps are repeated until we arrive to a solution P with $\sum_p w(p)\delta(p,X) \leq (1+\epsilon)\mathsf{OPT}$.

In each step, with probability $\Omega(\epsilon/k)$, the algorithm chooses an $\epsilon/3$-witness p and a center κ that is consistent with some hypothetical optimum solution O. However, it is not clear how to bound the running time of the algorithm. It can happen that the requests arriving to Q_κ have smaller and smaller radii. As we have seen for WEIGHTED k-CENTER, in such a scenario we cannot bound the number of steps even in $\mathbb{R}^1$ It is crucial to have some control on the sequence of radii that appear in the requests. Therefore, next we show how to ensure that the radii in the requests to center κ stay within a bounded range.

d) Initial Upper Bounds: For each point p, we compute a weak upper bound $u(p) \geq \delta(p,O)$ on the distance to the optimum solution. Then instead of starting with an arbitrary set of k centers, we bootstrap the algorithm by a solution approximately satisfying all these upper bounds. We argue that this can be done in such a way that ensures that the radii appearing in the requests to each center κ stay within a bounded range.

If a point p^* has weight $w(p^*)$, then $u(p^*) = \mathsf{OPT}/w(p^*)$ is an obvious upper bound on the distance of p^* to O: otherwise, we would have $\sum_p w(p)\delta(p,O) \geq w(p^*)\delta(p^*,O) > \mathsf{OPT}$. This bound was sufficient for the WEIGHTED k-CENTER problem, but the nature of WEIGHTED k-MEDIAN allows us to get much stronger upper bounds in many cases. For example, if there are c points of the same weight w roughly at the same position, then each of them should be at distance at most $\mathsf{OPT}/(wc)$ from O. Indeed, otherwise the total contribution of these c points to the sum would be greater than OPT. More generally, if there is a radius r such that total weight of the points at distance at most r from p is at least OPT/r, then we claim that p is at most distance $2r$ from O. Indeed, otherwise all these points would be at distance more than r from O, making their total contribution greater than OPT. Therefore, we can define $u(p) = 2r$, where r is the smallest radius with the property that the total weight of the points at distance at most r from p is at least OPT/r. Note that $u(p)$ can be determined in polynomial time from the weights of the points and their distance matrix.

Similarly to our WEIGHTED k-CENTER algorithm, we start with a 3-approximation of the constraints given by the upper bounds $u(p)$ for $p \in P$. Let us go through the points in a nondecreasing order of $u(p)$ and let us greedily choose a maximal independent set of the balls $\mathsf{ball}(p,u(p))$. We should find at most k such balls. Let us choose a center in each ball; it is easy to see that every point p has a selected center at distance at most $3u(p)$ from it. If center x_κ was selected to be a center in $\mathsf{ball}(p,u(p))$, then we initialize Q_κ with the request $(p,u(p))$. This ensures that during every step of the algorithm, it remains true that every point p is at distance at most $3u(p)$ from the current solution.

We run the algorithm for WEIGHTED k-MEDIAN with this initial solution. Before analyzing the algorithm, let us make a nontrivial change in the random selection. We have seen that with probability $\Omega(\epsilon/k)$, we select a random point p and $\kappa \in [k]$ such that $\delta(p,X) \geq (1+\epsilon/3)\delta(p,O)$ for some optimal solution O. A key claim of the proof is that with probability $\Omega(\epsilon/k)$, it is also true that $u(p) \leq 2k\delta(p,X)/\epsilon$ (see Lemma V.10). Intuitively, the total contribution of the $\epsilon/3$-witnesses that are too close to some center $x_\kappa \in X$ cannot be very large, because then all of these witnesses would be in a small ball, implying that the upper bound $u(p)$ should be smaller. Note that this is the point in the proof where we crucially utilize the exact definition of $u(p)$. With this claim at hand, we can modify the algorithm such that we are randomly choosing a point p satisfying $u(p) \leq 2k\delta(p,X)/\epsilon$, with probability proportional to $w(p)\delta(p,X)$. It remains true that p is an $\epsilon/3$-witness with probability $\Omega(\epsilon/k)$.

Let us analyze now the algorithm and bound the number of times a center x_κ is updated. We want to argue that the radius in the requests remains in a bounded range. Suppose that we update cluster κ with requests (p,r) and (p',r') (in either order) such that $r' \ll \epsilon^2 r/k$. If the algorithm does not fail, then there is a center x_κ satisfying both requests. By the triangle inequality, this means that the $\delta(p,p') \leq r + r' < r + \epsilon r/6$. Furthermore, by the constraint $u(p') \leq 2k\delta(p',X)/\epsilon = 2k(1+\epsilon/3)r'/\epsilon$ on our selection of the random point p', we have that $u(p')$ is much smaller than $\epsilon r/18$. At every step of the algorithm, the upper bound $u(p')$ is 3-approximately satisfied by the current solution X. Thus there should be a center in X much closer than $\epsilon r/6$ to p'. Together with $\delta(p,p') < r+\epsilon r/6$, it follows that there is always a center in X at distance at most $(1+\epsilon/3)r$ from p, contradicting the need for the request (p,r).

Thus the combination of the two facts that (1) the upper bounds are always satisfied approximately and that (2) the radius in the request is not much smaller than the upper bound implies that the radius in the requests stays within a bounded range. Then we can argue as in the case of the WEIGHTED k-CENTER problem. If every weight is rounded to a power of $(1+\epsilon)$, then each cluster is given requests with only a bounded number of different radii. If many requests arrive, then there is a long subsequence of the requests with the same radius. This means that the bound on the ϵ-scatter dimension can be used to bound the length of this subsequence, and hence the total number of requests to all clusters.

e) From Weighted k-Median *to General Norms Using Subgradients:* Next we show how to solve the clustering problem for an arbitrary monotone norm by interpreting it as collection of Weighted k-Median instances that we need to satisfy simultaneously. We will repeatedly solve such Weighted k-Median instances that are dynamically discovered during the course of the algorithm.

It will be convenient to use the notion of subgradients. For our purposes, it is sufficient to discuss subgradients in the context of a monotone norm $f\colon \mathbb{R}^n \to \mathbb{R}$. We say that $\boldsymbol{g}$ is a *subgradient* of f at point $\boldsymbol{x}$ if $f(\boldsymbol{x}) = \boldsymbol{g}^\intercal \boldsymbol{x}$ and $f(\boldsymbol{y}) \geq \boldsymbol{g}^\intercal \boldsymbol{y}$ for every $\boldsymbol{y} \in \mathbb{R}^n$. It is known that every monotone norm has a nonnegative subgradient $\boldsymbol{g} \geq 0$ at every point $\boldsymbol{x} \geq 0$. Checking whether a vector $\boldsymbol{g}$ is a subgradient at $\boldsymbol{x}$ and finding a subgradient at $\boldsymbol{x}$ can be formulated as convex optimization problems, hence can be (approximately) solved using the ellipsoid method if f can be efficiently computed (or approximated) [42].

Suppose that we have a current solution X and let $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ be the vector representing the distances of the points in P to X. Suppose that X is not (approximately) optimal: $f(\boldsymbol{x}) > (1+\epsilon)\mathsf{OPT}$. Let us compute a sugradient $\boldsymbol{g}$ of f at $\boldsymbol{x}$; we have $\boldsymbol{g}^\intercal \boldsymbol{x} = f(x) > (1+\epsilon)\mathsf{OPT}$ and $\boldsymbol{g}^\intercal \boldsymbol{y} \leq f(\boldsymbol{y}) = \mathsf{OPT}$ for the optimum solution $\boldsymbol{y}$. That is, $\boldsymbol{g}^\intercal \boldsymbol{x} \leq \mathsf{OPT}$ is a linear constraint satisfied by the optimum solution and violated by the current solution. Then defining the weights $w(p)$ based on the coordinates of $\boldsymbol{g}$ gives an instance of Weighted k-Median, with $\sum_p w(p)\delta(p,X) > (1+\epsilon)\mathsf{OPT}$ for the current solution X. Now we can proceed as above for the Weighted k-Median problem: we randomly choose a point p and cluster κ, introduce a new request into Q_κ, find a new center x_κ, etc., until we arrive to a solution X with $\sum_p w(p)\delta(p,X) \leq (1+\epsilon)\mathsf{OPT}$. If this new solution X is still nonoptimal for the original norm problem, that is, $f(\boldsymbol{x}) > (1+\epsilon)\mathsf{OPT}$, then we can again compute a subgradient, find a violated linear constraint (possibly the same as in the previous step). We repeat this until we find a solution with $f(\boldsymbol{x}) \leq (1+\epsilon)\mathsf{OPT}$.

Defining the upper bounds and bootstrapping the algorithm with a solution approximately satisfying the upper bounds were crucial for the analysis of the Weighted k-Median algorithm. For general norms, we can again define the upper bounds once we have the weights w based on the violated linear constraint $\boldsymbol{g}^\intercal \boldsymbol{x} \leq \mathsf{OPT}$. However, these upper bounds would not be useful for the analysis, as they would depend on the violated linear constraint, hence would change during the algorithm.

Intuitively, we can see the constraint $f(\boldsymbol{x}) \leq \mathsf{OPT}$ as an infinite number of Weighted k-Median instances, corresponding to the linear constraints $\boldsymbol{g}^\intercal \boldsymbol{x} \leq \mathsf{OPT}$ for *every* subgradient $\boldsymbol{g}$ of f. We would like to define $u(p)$ to be the smallest possible upper bound that can be assigned to p among all of these infinitely many Weighted k-Median instances. Determining this value seems to be a difficult task, but actually the answer is very simple. Recall that $u(p)$ was defined as twice the smallest r such that $\mathsf{ball}(p,r)$ contains total weight at least OPT/r. Thus to define the upper bound $u(p)$, we need to know what the maximum weight of the points in $\mathsf{ball}(p,r)$ can be among the infinitely many instances corresponding to all the subgradients. Let $\boldsymbol{b}$ be the characteristic vector of $\mathsf{ball}(p,r)$ (i.e., every coordinate is 1 or 0, depending on whether a point is in or not in the ball). Then the question is to determine the maximum of $\boldsymbol{g}^\intercal \boldsymbol{b}$ among all subgradients $\boldsymbol{g}$. It is easy to see that this maximum is exactly $f(\boldsymbol{b})$: if $\boldsymbol{g}$ is a subgradient at $\boldsymbol{b}$, then $\boldsymbol{g}^\intercal \boldsymbol{b} = f(\boldsymbol{b})$; if $\boldsymbol{g}$ is a subgradient at an arbitrary point $\boldsymbol{y}$, then $\boldsymbol{g}^\intercal \boldsymbol{b} \leq f(\boldsymbol{b})$. Thus we can determine the maximum weight of any ball and define the upper bounds accordingly. With these definitions, the analysis of the Weighted k-Median algorithm go through for general mononote norms. The two main properties of the upper bounds remain valid: (1) the upper bounds are satisfied by the optimum solution and (2) we can restrict our random choice of p to points where the distance to the solution is not much smaller than $u(p)$.

In summary, the final algorithm consists of the following steps (see Figure 3). First we compute the upper bounds $u(p)$ and greedily find a 3-approximate solution satisfying these constraints. Then we repeat the following steps until we reach a solution X for which the distance vector $\boldsymbol{x}$ satisfies $f(\boldsymbol{x}) \leq (1+\epsilon)\mathsf{OPT}$. We compute a subgradient $\boldsymbol{g}$ of f at $\boldsymbol{x}$ to obtain a violated linear constraint $\boldsymbol{g}^\intercal \boldsymbol{x} \leq \mathsf{OPT}$. We randomly choose a point p (according to the distribution described above) and require that p be at most distance $\delta(p,X)/(1+\epsilon/3)$ from the solution, that is, we obtain a violated distance constraint. Then we randomly choose a cluster $\kappa \in [k]$ and require that this distance constraint be satisfied by center x_κ. Thus we put the request $(p, \delta(p,X)/(1+\epsilon/3)$ into Q_κ find a new x_κ that satisfy the cluster constraints imposed by the requests in Q_κ, if possible. We repeat these steps until we arrive to a solution X with distance vector $\boldsymbol{x}$ satisfying $f(\boldsymbol{x}) \leq (1+\epsilon)\mathsf{OPT}$. Our analysis shows that each step is consistent with a hypothetical optimum solution O with probability $\Omega(\epsilon/k)$. Moreover, if ϵ-scatter dimension is bounded, then the algorithm has to find a solution or fail after a bounded number of iterations.

f) (Algorithmic) ϵ-Scatter Dimension: After the general algorithm capable of handling any monotone norm objective, our second main contribution is bounding the ϵ-scatter dimension of various classes of metrics (Section VI). In the interest of space, we do not go into the details of these (mostly combinatorial) proofs, but give only a brief overview.

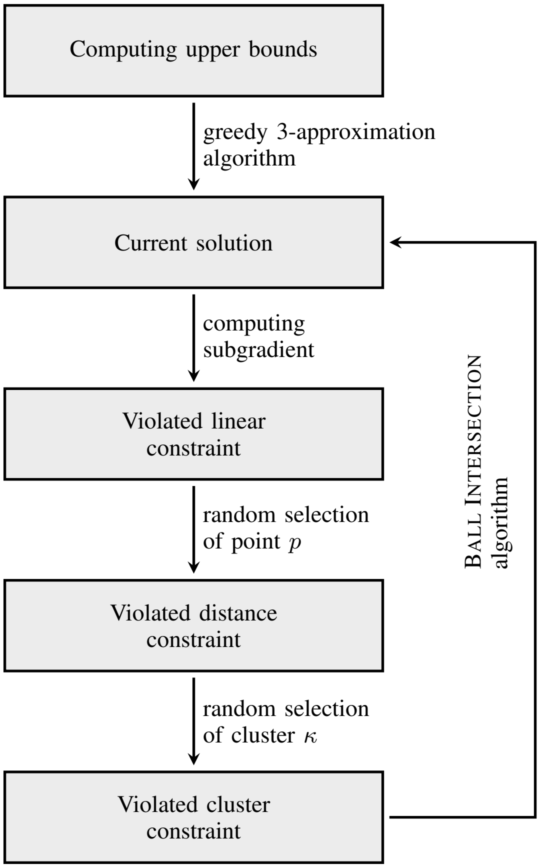
Computing upper bounds
greedy 3-approximation algorithm
Current solution
computing subgradient
Violated linear constraint
random selection of point p
Violated distance constraint
random selection of cluster κ
Violated cluster constraint
Ball Intersection algorithm

Fig. 3: Overall structure of the main algorithm.

- **Bounded Doubling Dimension.** As outlined in the introduction, the set of points as well as the set of centers in an ϵ-scattering both form an ϵ-packing of a unit ball implying that any metric of doubling dimension d has ϵ-scatter dimension $(1/\epsilon)^{\mathcal{O}(d)}$. See Theorem I.3.
- **Bounded-Treewidth Graph Metrics.** The ϵ-scatter dimension bound for metrics defined by the shortest path metric of bounded-treewidth graphs is obtained by a delicate combinatorial proof that exploits both structure of the graph and properties of the ϵ-scattering. The bound we obtain is $\mathtt{tw}^{1/\epsilon^{\mathcal{O}(\mathtt{tw})}}$ for graphs of treewidth $\mathtt{tw}$, that is, double exponential in $\mathtt{tw}$ for fixed ϵ. It remains is an interesting open question if this bound can be improved.
- **Planar Graph Metrics.** As outlined in the introduction, we can employ a known metric embedding result to reduce the problem of bounding the ϵ-scatter dimension of planar graphs to bounding the ϵ-scatter dimension of bounded-treewidth graphs. In particular, the result by Fox-Epstein, Klein, and Schild [16] provides an (approximate) metric embedding of planar metrics into low-treewidth metrics, which can be used to obtain a $2^{2^{\mathsf{poly}(1/\epsilon)}}$ bound on the ϵ-scatter dimension of planar graph metrics.
- **Continuous High-Dimensional Euclidean Space.** As mentioned in the introduction, the high-dimensional Euclidean space does not have bounded ϵ-scatter dimension. However, in the continuous Euclidean space, where any point of the space can be a center, we can bound the *algorithmic ϵ-scatter dimension*. Towards this, we replace the center player by an algorithmic "player" applying the algorithm by Kumar and Yildirim [43] for Weighted 1-Center. To achieve bounded algorithmic ϵ-scatter dimension, this algorithm would require, however, a bounded aspect ratio of the radii in the input requests. We therefore prove an aspect-ratio condition (which holds even for general metrics) implying that it is sufficient for the algorithm to handle instances with aspect-ratio $\mathcal{O}(1/\epsilon)$. We combine this result with the algorithm by Kumar and Yildirim to prove bounded algorithmic ϵ-scatter dimension for continuous high-dimensional Euclidean space, that is, Theorem I.7.

III. Preliminaries

a) Classes of Metric Clustering Spaces: A *metric clustering space* (or metric space for brevity) is a triple $M = (P, F, \delta)$ where P is a finite set of n *data points*, F is a (possibly infinite) set of potential locations of cluster centers, and δ is a metric on $P \cup F$. Sets P and F are not necessarily disjoint. (For example, it is natural for clustering problems to have $P = F$ or $P \subseteq F$.) Given any point $u \in P \cup F$ in the metric space and a radius $r \in \mathbb{R}_+$, we denote by $\mathsf{ball}_\delta(u, r) = \{\, v \in P \cup F \mid \delta(u, v) \le r \,\}$ the ball of radius r centered around u. We drop the subscript δ if the distance function is clear from the context.

By $|M|$ we denote the space needed to represent the metric space M in the memory. If M is finite then $|M|$ is polynomial in $|F|$, $|P|$ and the space needed for storing a point and a center, respectively. If F is infinite (for example, in the continuous Euclidean setting, $F = \mathbb{R}^d$), $|M|$ is polynomial in $|P|$ and the space of storing a point.

A *class* $\mathcal{M}$ of metric spaces is a (infinite) set of metric spaces. This paper focuses on metric classes that are closed under scaling distances by a constant. We consider the following classes of metric clustering spaces:

- **Graph Metric:** In the case of graph metric, we are given a (weighted) graph $G = (V, E)$ and the metric δ_G on V as the shortest path metric, i.e., $\delta_G(u, v)$ is the shortest distance of a path connecting u and v. The clustering space (P, F, δ_G) is given such that $P, F \subseteq V$.

- **Continuous Euclidean Spaces:** In this case, we are allowed to choose centers from the (high-dimensional) continuous Euclidean space $F = \mathbb{R}^d$. The set $P \subsetneq \mathbb{R}^d$ is a finite set of points.
- **Doubling Metric:** The *doubling dimension* of a metric space (X, δ), denoted as d, is the smallest $m > 0$ such that every ball of radius r in the metric can be covered by 2^m balls of radius $\frac{r}{2}$. Note that a d-dimensional Euclidean metric has doubling dimension $\mathcal{O}(d)$.

b) Treewidth: A *tree decomposition* of a graph G is a pair (T, β) where T is a tree, $\beta\colon V(T) \to 2^{V(G)}$, $V(T)$ and $V(G)$ denote the vertices of the tree T and G respectively, with the following properties.

1) For each $v \in V(G)$, there exists $t \in V(T)$ such that $v \in \beta(t)$,
2) For each $(u, v) \in E(G)$, there exists $t \in V(T)$ such that $u, v \in \beta(t)$, and
3) For each $v \in V(T)$, the subgraph induced by T on $\{t : v \in \beta(t)\}$, is connected.

The *width* of the tree decomposition (T, β) is $\max_{t \in V(T)} |\beta(t)| - 1$. The *treewidth* of a graph G is the minimum width over all tree decompositions of G.

Subgradients of Norms: We state definitions and summarize basic facts about subgradients of norms that we will use throughout the paper.

Fact III.1. *Any norm is a convex function.*

Definition III.2 (Subgradient)**.** *A subgradient of a convex function $f\colon \mathbb{R}^n \to \mathbb{R}$ at any point $\boldsymbol{x} \in \mathbb{R}^n$ is any $\boldsymbol{g} \in \mathbb{R}^n$ such that the following holds for every $\boldsymbol{y} \in \mathbb{R}^n$*

$$f(\boldsymbol{y}) \geq f(\boldsymbol{x}) + \boldsymbol{g}^{\intercal}(\boldsymbol{y} - \boldsymbol{x});$$

we denote by $\partial f(\boldsymbol{x})$ the set of subgradients of f at $\boldsymbol{x}$.

The following fact summarizes various useful properties of subgradients specialized to norm functions. Because we are apply norm objectives exclusively to non-negative distance vectors, we call (slightly abusing terminology) a restriction of a norm to $\mathbb{R}^n_{\geq 0}$ a norm as well.

Fact III.3 ([17])**.** *Let $f\colon \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a norm and $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$. If $\boldsymbol{g}$ is a subgradient of f at $\boldsymbol{x}$, then $f(x) = \boldsymbol{g}^{\intercal}\boldsymbol{x}$ and $f(y) \geq \boldsymbol{g}^{\intercal}\boldsymbol{y}$ for all $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$. Further, if f is monotone, there exists a subgradient $\boldsymbol{g} \in \partial f(\boldsymbol{x})$ such that $\boldsymbol{g} \geq 0$.*

The following observation is an immediate consequence of Fact III.3.

Observation III.4. *Let $\partial f = \bigcup_{\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}} \partial f(\boldsymbol{y})$ be the set of all subgradients of f. Then for any $\boldsymbol{x} \in \mathbb{R}^n_{\geq 0}$, we have that*

$$f(\boldsymbol{x}) = \max_{\boldsymbol{g} \in \partial f} \boldsymbol{g}^{\intercal}\boldsymbol{x}\,.$$

Definition III.5 (ϵ-Approximate Subgradient)**.** *Let $f\colon \mathbb{R}^n_{\geq 0} \to \mathbb{R}_{\geq 0}$ be a norm and let $\epsilon > 0$. We define the set $\partial_\epsilon f(\boldsymbol{x})$ of ϵ-*approximate subgradients of f at *$\boldsymbol{x}$ to contain all $\boldsymbol{g} \in \mathbb{R}^n_{\geq 0}$ such that the following two conditions hold*

(i) $f(\boldsymbol{y}) \geq \boldsymbol{g}^{\intercal}\boldsymbol{y}$ for each $\boldsymbol{y} \in \mathbb{R}^n_{\geq 0}$, and
(ii) $f(\boldsymbol{x}) \leq (1 + \epsilon)\boldsymbol{g}^{\intercal}\boldsymbol{x}$.

It is known that approximate subgradients of convex functions can be computed efficiently via an (approximate) value oracle for the function through reductions shown by Grötschel, Lovasz and Schrijver in their classic book [42]. While the reduction in [42] appears to take at least $\Omega(n^{10})$ calls to the oracle, there exist faster methods assuming additional properties of the convex function, for example, see [44], [45]. Specifically for ℓ_p norms, closed formulas describing the sets of subgradients are known and used in practice.

Some Terminology and Notation: Let $M = (P, F, \delta)$ be a clustering space on $n = |P|$ data points. Let $\boldsymbol{b} \in \mathbb{R}^P_{\geq 0}$ be an n-dimensional vector. We interpret $\boldsymbol{b}$ as assigning each point $p \in P$ a non-negative value denoted $b(p)$. That is, $\boldsymbol{b} = (b(p))_{p \in P}$. For example, given a subset $X \subseteq F$ of centers, we define the *distance vector* $\boldsymbol{\delta}(P, X) = (\delta(p, X))_{p \in P}$. If $B \subseteq P$ is a subset of points then $\mathbf{1}_B \in \{0, 1\}^P$ denotes the *characteristic vector of* B, that is, it assigns value 1 to any $b \in B$ and 0 to any $p \in P \setminus B$. If $p \in P$ and $\alpha \geq 0$ then we denote by $\mathbf{1}_{p,\alpha}$ the binary vector $\mathbf{1}_{\mathsf{ball}(p,\alpha) \cap P}$.

IV. ϵ-Scatter Dimension

In this section, we introduce the concept of ϵ-scatter dimension formally, which plays a central role in our algorithmic framework. The following definition is a formalization of the "center-point game" presented in the introduction.

Definition IV.1 (ϵ-Scatter Dimension)**.** *We are given a class $\mathcal{M}$ of finite metric spaces, a space $M = (P, F, \delta)$ in $\mathcal{M}$, and some $\epsilon \in (0, 1)$. An ϵ-*scattering in M *is a sequence $(x_1, p_1) \ldots, (x_\ell, p_\ell)$ of center-point pairs $x_i \in F$, $p_i \in P$, $i \in [\ell]$ such that*

$$\delta(x_i, p_j) \leq 1 \qquad \text{for all } 1 \leq j < i \leq \ell \qquad \text{(covering)}$$
$$\delta(x_i, p_i) > 1 + \epsilon \qquad \text{for all } i \in [\ell] \qquad (\epsilon\text{-refutation})$$

*The ϵ-*scatter dimension of M *is the maximum length of an ϵ-scattering in it. The ϵ-*scatter dimension of $\mathcal{M}$ *is the supremum of the ϵ-scatter dimension over all $M \in \mathcal{M}$.*

Note that for any ϵ-scattering $(x_1, p_1), \ldots, (x_\ell, p_\ell)$, any subsequence $(x_{i_1}, p_{i_1}), \ldots, (x_{i_{\ell'}}, p_{i_{\ell'}})$ where $i_1 < \cdots < i_{\ell'}$ and $\ell', i_j \in [\ell]$, $j \in [\ell']$ is an ϵ-scattering as well.

As described in Theorem I.8, we show that bounded (algorithmic) ϵ-scatter dimension is essentially sufficient to yield an EPAS for Norm k-Clustering in the respective metric space. In Section VI-B we show that bounded treewidth and planar graph metrics, and bounded doubling metrics have bounded ϵ-scatter dimension. This allows us to obtain EPASes in all these metrics. To handle high-dimensional Euclidean space, we resort to an algorithmic version of ϵ-scatter dimension.

Optimizing the Centering Strategy: Recall the example from the introduction showing that the ϵ-scatter dimension of the high-dimensional (continuous) Euclidean space $\mathbb{R}^d$ is be unbounded. We constructed an ϵ-scattering $(x_1, p_1), \ldots, (x_{d-1}, p_{d-1})$ where x_i is the i-th unit vector scaled by $1/\sqrt{2}$ and where $p_i = -x_i$ for all $i \in [d-1]$. Note that in this example the unit ball around the origin contains all the points in the sequence. Hence, the above example would collapse if the center player would improve her strategy. This motivates us to consider a variant where we replace the center player with an algorithm that computes centers more prudently. Further, employing algorithms allows us also to handle infinite spaces.

Ball Intersection *Problem and Algorithm:* Towards this, we formalize the algorithmic problem the center player has to solve. We adopt and generalize a dual interpretation of the center-point game in which the center player is trying to find a center in the *intersection* of all unit balls around the points played by the point player. In fact, we consider the more general setting of non-uniform balls where each point p in the scattering has its own dedicated radius r.

Let $\mathcal{M}$ be a class of metric spaces (P, F, δ) with possibly infinite center sets F. We define the following search problem.

Ball Intersection
Input: A metric space $M = (P, F, \delta) \in \mathcal{M}$, a set finite set $Q \subsetneq P \times \mathbb{R}_+$ of *distance constraints*.
Output: A point $x \in F$ *satisfying* all distance constraints, that is, $\delta(x, p) \leq r$ for each $(p, r) \in Q$, if such a point exists and "fail" otherwise.

For finite metric spaces, the Ball Intersection problem can be solved efficiently by exhaustively searching the center space F. Unfortunately, we are not aware of exact algorithms for Ball Intersection for certain infinite metric spaces such as high-dimensional continuous Euclidean space. We therefore work with approximate algorithms. To define this formally, we say that a center $x \in F$ *η-satisfies* the distance constraint $(p, r) \in Q$ for some error parameter $\eta > 0$, if $\delta(x, p) \leq (1+\eta)r$. Let $\mathcal{C}_\mathcal{M}$ be a (deterministic) algorithm whose input is an instance of Ball Intersection and an error parameter $\eta > 0$. The algorithm is called an *approximate* Ball Intersection *algorithm* (or Ball Intersection algorithm for short) if it satisfies the following conditions.

(i) The algorithm outputs a center that η-satisfies all distance constraints or it fails.
(ii) If there exists a center satisfying all points distance constraints exactly, then the algorithm does not fail.
(iii) The running time of $\mathcal{C}_\mathcal{M}$ is $\mathsf{poly}(|M|, 1/\eta)$.

We remark that there is an approximate Ball Intersection algorithm for high-dimensional Euclidean space [43], which we employ in Section VI-D to prove bounded algorithmic ϵ-scatter dimension of this metric.

Algorithmic ϵ-Scatter Dimension: The definition of *algorithmic ϵ-scatter dimension* is based on the notion of *$(\mathcal{C}_\mathcal{M}, \epsilon)$-scattering*, which is a variant of ϵ-scattering: Centers are chosen via an (approximate) Ball Intersection algorithm $\mathcal{C}_\mathcal{M}$ rather than by an adversarial center-player. Intuitively, we maintain a dynamic instance of Ball Intersection that is augmented by adding distance constraints (p, r) one by one. In the context of $(\mathcal{C}_\mathcal{M}, \epsilon)$-scattering, we call the distance constraints (p, r) *requests*, which are satisfied by the Ball Intersection algorithm sequentially.

Definition IV.2 (Algorithmic ϵ-Scatter Dimension)**.** *Let $\mathcal{M}$ be a class of metric spaces with* Ball Intersection *algorithm $\mathcal{C}_\mathcal{M}$, let $M = (P, F, \delta)$ be a metric in $\mathcal{M}$, and let $\epsilon \in (0, 1)$ Moreover, let $p_i \in P$, $x_i \in F$, and $r_i \in \mathbb{R}_+$ for each $i \in [\ell]$ where ℓ is a positive integer. The sequence $(x_1, p_1, r_1), \ldots, (x_\ell, p_\ell, r_\ell)$ is called an* $(\mathcal{C}_\mathcal{M}, \epsilon)$-scattering *if the following two conditions hold.*

(i) We have $x_i = \mathcal{C}_\mathcal{M}(M, \{(p_1, r_1), \ldots, (p_{i-1}, r_{i-1})\}, \epsilon/2)$ for each $2 \leq i \leq \ell$. (There is no requirement regarding the first center x_1 in the sequence.)
(ii) Moreover, $\delta(x_i, p_i) > (1+\epsilon)r_i$ for each $i \in [\ell]$.

We say that $\mathcal{M}$ has algorithmic $(\epsilon, \mathcal{C}_\mathcal{M})$-scatter dimension *$\lambda_\mathcal{M}(\epsilon)$ if any $(\mathcal{C}_\mathcal{M}, \epsilon)$-scattering contains at most $\lambda_\mathcal{M}(\epsilon)$ many triples with the same radius value. The* algorithmic ϵ-scatter dimension *of $\mathcal{M}$ is the minimum algorithmic $(\epsilon, \mathcal{C}_\mathcal{M})$-scatter dimension over any* Ball Intersection *algorithm $\mathcal{C}_\mathcal{M}$ for $\mathcal{M}$.*

When the family $\mathcal{M}$ is clear from the context we drop the subscript $\mathcal{M}$ from $\lambda_\mathcal{M}(\epsilon)$ and $\mathcal{C}_\mathcal{M}$. Note that, in contrast to the ϵ-scatter dimension, for algorithmic ϵ-scatter dimension we demand that the number of triples per radius value be bounded rather than the total length of the sequence. In fact, this stronger requirement would not hold for high-dimensional Euclidean spaces whereas the weaker (algorithmic) requirement turns out to be sufficient for our results. Another noteworthy difference is that a subsequence of an $(\mathcal{C}_\mathcal{M}, \epsilon)$-scattering is not necessarily a $(\mathcal{C}_\mathcal{M}, \epsilon)$-scattering itself because it may not be consistent with the behavior of algorithm $\mathcal{C}_\mathcal{M}$.

Relation Between Algorithmic and non-Algorithmic ϵ-Scatter Dimension: The following lemma shows that the algorithmic ϵ-scatter dimension indeed generalizes the ϵ-scatter dimension for finite metric spaces.

Lemma IV.3. *Any class of finite, explicitly given, metric spaces with ϵ-scatter dimension $\lambda(\epsilon)$ has also algorithmic ϵ-scatter dimension $\lambda(\epsilon)$.*

Proof. Let $M = (P, F, \delta)$ be a metric space in the given class along with a set Q of distance constraints. Our BALL INTERSECTION algorithm exhaustively searches F to find a center x satisfying all distance constraints *exactly*. If no such point exists the algorithm fails. Let $\mathcal{C}$ denote this algorithm. Consider any $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-scattering. Notice that any sub-sequence of triples with the same radius value forms an ϵ-scattering. Hence the sequence contains at most $\lambda(\epsilon)$ many triples for any radius value. □

Aspect-Ratio Lemma for Algorithmic ϵ-Scatter Dimension: The following is a handy consequence of bounded algorithmic ϵ-scatter dimension that we use in proving our result. It strengthens the properties of an $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-scattering by bounding the number of triples whose radii lie in an interval of bounded aspect-ratio (rather than bounding the number of triples with the same radius value).

Lemma IV.4. *Let $\mathcal{M}$ be a class of metric spaces of algorithmic ϵ-scatter dimension $\lambda(\epsilon)$. Then there exists a* BALL INTERSECTION *algorithm $\mathcal{C}_{\mathcal{M}}$ with the following property. Given $\epsilon \in (0,1)$, $a > 0$, and $\tau \geq 2$, any $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-scattering contains $\mathcal{O}(\lambda(\epsilon/2)(\log \tau)/\epsilon)$ many triples whose radii lie in the interval $[a, \tau a]$.*

Proof. It suffices to show the weaker claim that the number of requests in the interval $[a, (1 + \epsilon/100)a]$ is at most $2\lambda(\epsilon/2)$. This claim implies the lemma because the interval $[a, \tau a]$ can be covered with $\mathcal{O}((\log \tau)/\epsilon)$ many intervals of the form $[(1+\epsilon/100)^j, (1+\epsilon/100)^{j+1}]$, $j \in \mathbb{Z}$.

Let $\mathcal{A}_{\mathcal{M}}$ be an BALL INTERSECTION algorithm such that the algorithmic $(\epsilon, \mathcal{A}_{\mathcal{M}})$-scatter dimension is $\lambda(\epsilon)$. Let $\eta \in (0,1)$ be the input error parameter. Consider the BALL INTERSECTION algorithm $\mathcal{C}_{\mathcal{M}}$ that works as follows. For any of the input requests (p, r) we round r to r', which is the smallest power of $1+\eta/50$ larger than r. We then invoke $\mathcal{A}_{\mathcal{M}}$ on the rounded requests with error parameter $\eta/2$ and output the center returned by $\mathcal{A}_{\mathcal{M}}$. Clearly, this algorithm is an $(1+\eta)$-approximate BALL INTERSECTION algorithm (for the original requests).

Consider any algorithmic $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-scattering $(x_1, p_1, r_1), \ldots, (x_\ell, p_\ell, r_\ell)$. Let r'_i, $i \in [\ell]$ be the rounded radii computed by $\mathcal{C}_{\mathcal{M}}$. Let $\epsilon' = \epsilon/2$. Let $1 \leq j < i \leq \ell$. We have $\delta(x_i, p_i) > (1+\epsilon) r_i \geq (1+\epsilon)/(1+\epsilon/100) r'_i \geq (1+\epsilon') r'_i$. Moreover, we have $\delta(x_i, p_j) \leq (1 + \epsilon'/2) r'_i$. Hence, the sequence $(x_1, p_1, r'_1), \ldots, (x_\ell, p_\ell, r'_\ell)$ is an algorithmic $(\mathcal{C}_{\mathcal{M}}, \epsilon')$-scattering. The radii in the requests (p_i, r_i), $i \in [\ell]$ that lie in the interval $[a, (1 + \epsilon/100)a]$ are rounded by $\mathcal{C}_{\mathcal{M}}$ to at most two distinct radius values because $\mathcal{C}_{\mathcal{M}}$ is invoked with error parameter $\eta = \epsilon/2$. Hence the (unrounded) sequence contains at most $2\lambda(\epsilon') = 2\lambda(\epsilon/2)$ many triples with radii in the interval $[a, (1 + \epsilon/100)a]$. This completes the proof of the claim and therefore of the lemma. □

V. FRAMEWORK FOR EFFICIENT PARAMETERIZED APPROXIMATION SCHEMES

Main Result: We are now ready to state our main result. In the remainder of this section, we prove the following theorem, restated from the introduction. In Section V-A, we describe the EPAS and give some intuition. In Section V-B, we give a full, technical analysis.

Theorem I.8. *Let $\mathcal{M}$ be a class of metric spaces that is closed under scaling distances by a positive constant. There is a randomized algorithm that computes for any* NORM k-CLUSTERING *instance $\mathcal{I} = (M, f, k)$ with metric $M = (P, F, \delta) \in \mathcal{M}$, and any $\epsilon \in (0,1)$, with high probability a $(1 + \epsilon)$-approximate solution if the following two conditions are met.*

(i) There is an efficient algorithm evaluating for any distance vector $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ the objective $f(\boldsymbol{x})$ in time $T(f)$.

(ii) There exists a function $\lambda\colon \mathbb{R}_+ \to \mathbb{R}_+$, such that for all $\epsilon > 0$, the algorithmic ϵ-scatter dimension of $\mathcal{M}$ is at most $\lambda(\epsilon)$.

The running time of the algorithm is $\exp\left(\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right) \cdot \textbf{poly}(|M|) \cdot T(f)$.

A. Algorithm

Our algorithm is stated formally in Algorithm 1. We informally summarize the key steps of our algorithm, which we also outlined partially in the technical overview. We also give some intuition of the analysis.

Using standard enumeration techniques, we assume that we know (a sufficiently exact approximation of) the optimum objective function value OPT. Our goal is to satisfy the *convex constraint* $f(\boldsymbol{x}) \leq (1+\epsilon)\mathsf{OPT}$ imposed on the distance vector $\boldsymbol{x} \in \mathbb{R}^P_{\geq 0}$ (which represents the distance vector $\boldsymbol{\delta}(P, X)$ induced by the feasible solution $X \subseteq F$). By Observation III.4, this constraint is equivalent to (infinitely many) *linear constraints* $\boldsymbol{w}^\intercal \boldsymbol{x} \leq (1+\epsilon)\mathsf{OPT}$ where $\boldsymbol{w} \in \partial f$ is any subgradient of f.

To illustrate the main idea, we first describe a highly simplified, but failed attempt. We consider in each iteration of the while-loop (lines 8–15) a candidate solution X. If $f(\boldsymbol{x}) \leq (1+\epsilon)\mathsf{OPT}$, then we are done. Otherwise, we compute an ($\epsilon/10$-approximate) subgradient $\boldsymbol{w}$ of f at $\boldsymbol{x}$ in line 9. Since $\boldsymbol{w}^\intercal \boldsymbol{x} = f(\boldsymbol{x}) > (1+\epsilon)\mathsf{OPT}$, this constitutes a *violated* linear constraint. Consider sampling a point $p \in P$ with probability proportional to its contribution $w(p)\delta(p, X)$ to the objective $f(\boldsymbol{x}) = \boldsymbol{w}^\intercal \boldsymbol{x}$ (line 11). An averaging argument shows that with probability $\Omega(\epsilon)$, the sampled point p satisfies $\delta(p, X) > (1 + \epsilon/3)\delta(p, O)$ for some fixed hypothetical optimum solution O. In this event, we identified a violated *distance constraint*, and call p an $\epsilon/3$-*witness* for X. We assign p to a cluster $\kappa \in [k]$ picked uniformly at random, which equals the correct cluster of p in O with probability $1/k$. Assuming that both events occur, this allows us to add the *request* (p, r) with radius value $r = \delta(p, X)/(1 + \epsilon/3)$ to the *cluster constraint* Q_κ imposed on the cluster with index κ. (See lines 13 and 14.) Here, we refer to the set Q_κ of requests for cluster κ as cluster constraint of κ.

Fix cluster index $\kappa \in [k]$. Let $(p_\kappa^{(1)}, r_\kappa^{(1)}), \ldots, (p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ be the sequence of requests added to the cluster constraint associated with cluster κ. Let $x_\kappa^{(i)}$, $i \in [\ell]$ be the center of cluster κ just before adding the request $(p_\kappa^{(i)}, r_\kappa^{(i)})$ to Q_κ. The key observation is that the sequence of triples $(x_\kappa^{(1)}, p_\kappa^{(1)}, r_\kappa^{(1)}), \ldots, (x_\kappa^{(\ell)}, p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ forms an algorithmic ϵ-scattering. We would like to argue that the length of this sequence is bounded because the algorithmic ϵ-scatter dimension is bounded. Unfortunately, the scatter dimension bounds only the number of triples per radius value but not the overall length of the sequence.

To address this issue, we compute in line 1 an initial upper bound $u(p)$ on the radius of any point $p \in P$. We (approximately) satisfy these initial distance constraints for all points in a greedy pre-processing step (see lines 2–7). We maintain the distance constraints during the main phase by adding them as initial requests (see line 5). The upper bound $u(p)$ is a rough estimate of the smallest radius r that may be imposed on p as part of any request (p, r). We modify the sampling process in the main phase (see line 11) to sample only from a subset of points whose distance to X is not much smaller than their initial upper bound $u(p)$. We show via a careful argument that every request (p, r) we make is consistent with O with probability $\Omega(\epsilon/k)$. We argue, moreover, that all radii of requests made for a particular cluster are within a factor $\mathcal{O}(k/\epsilon^2)$ of each other. The initial upper bounds are computed by detecting "dense" balls (line 1) in the input instance in the sense that they would receive high weight by some subgradient of the objective norm and would therefore require that any near-optimal solution must place a center in the vicinity of that dense ball.

B. Analysis

Overview: The analysis consists of establishing the following three facts. First, if the algorithm terminates without failure, it computes a $(1+\epsilon)$-approximation. Second, the algorithm terminates—with or without failure—after a number of iterations that depends on k and ϵ only. Third, the algorithm does not fail with a probability that depends only on k and ϵ as well.

Algorithm 1: Framework for NORM k-CLUSTERING

```
Data: Instance
      $\mathcal{I} = ((P, F, \delta), k, f\colon \mathbb{R}^P_{\geq 0} \to \mathbb{R}_{\geq 0})$ of
      NORM $k$-CLUSTERING, error parameter
      $\epsilon \in (0, 1)$, $\mathsf{OPT} > 0$, BALL
      INTERSECTION algorithm $\mathcal{C}$ according to
      Lemma IV.4
Result: Solution $X$ of cost at most $(1 + \epsilon)\mathsf{OPT}$
        if solution of cost at most $\mathsf{OPT}$ exists
1  For each $p \in P$, compute
     $u(p) = \min\{\, \alpha > 0 \mid f(\mathbf{1}_{p, \alpha/3}) \geq 3\mathsf{OPT}/\alpha \,\}$;
2  Sort $P$ in non-decreasing order of $u(p)$;
3  Mark $p_i \in P$ if $\mathsf{ball}(p_i, u(p_i))$ is disjoint from
     $\mathsf{ball}(p_j, u(p_j)$ for every $j < i$;
4  Let $p^{(1)}$, ..., $p^{(k')}$ be the marked points.;
     // Lemma V.5 shows that $k' \leq k$
5  Let $Q_\kappa = \{(p^{(\kappa)}, u(p^{(\kappa)}))\}$ for all $\kappa \in [k']$;
6  Let $Q_\kappa = \emptyset$ for all $\kappa$ with $k' < \kappa \leq k$;
7  Let $X = (x_1, \ldots, x_k)$ be any set of centers
     where $x_\kappa$ satisfies the requests in $Q_\kappa$;
8  while $f(\boldsymbol{\delta}(P, X)) > (1 + \epsilon)\mathit{OPT}$ do
9      $\boldsymbol{w} \leftarrow \epsilon/10$-subgradient of $f$ at $\boldsymbol{\delta}(P, X)$;
10     $A \leftarrow \left\{ p \in P \mid \delta(p, X) \geq \frac{\epsilon u(p)}{1000k} \right\}$;
11     Sample an element $p \in A$ where
         $\mathbf{Pr}\,[p = a] = \frac{w(a)\delta(a,X)}{\sum_{b \in A} w(b)\delta(b,X)}$ for any $a \in A$;
12     Pick cluster $\kappa \in [k]$ for $p$ uniformly at
         random;
13     $Q_\kappa \leftarrow Q_\kappa \cup \{(p, \delta(p, X)/(1 + \epsilon/3))\}$;
14     $x_\kappa \leftarrow \mathcal{C}(Q_\kappa, \epsilon/10)$ if no $x_i$ was found then
         fail ;
15 end
16 return $X$;
```

The first step of the analysis follows immediately from the stopping criterion (line 8) of the while loop.

Observation V.1 (Correctness). *If the algorithm terminates successfully (that is, without failure), then it outputs a $(1 + \epsilon)$-approximate solution.*

The second step of the analysis is summarized in the following lemma, which we prove in Subsection V-B1.

Lemma V.2 (Runtime bound). *The algorithm terminates after $\mathcal{O}\left(\frac{k(\log k/\epsilon)\lambda(\epsilon/10)}{\epsilon}\right)$ iterations—with or without failure.*

With these two insights at hand, we are left with the third step summarized by the following lemma, which we prove in Subsection V-B2.

Lemma V.3 (Probability bound)**.** *The algorithm terminates successfully (that is, without failure) with probability* $\exp\left(-\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right)$.

We repeat the algorithm $\exp\left(\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right)$ many times and hence succeed with high probability by Lemma V.3.

The remainder of this section is dedicated to proving Lemmas V.2 and V.3, thereby completing proof of the main Theorem I.8.

1) Bounding the Number of Iterations: In this subsection, we prove Lemma V.2. The proof consists in three steps. First, we argue that the initial upper distance bounds $u(p)$ that we compute for each point $p \in P$ are (i) consistent with any optimum solution (Lemma V.4), and (ii) approximately satisfied throughout the algorithm (Lemma V.5). Second, we establish that the radii in the requests made for any particular cluster are within a bounded factor (aspect ratio) of each other (Lemma V.6). The third step consists in proving that, for any particular cluster, the sequence of requests along with the corresponding centers constitute an algorithmic $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-scattering of bounded aspect ratio. Hence we can use Lemma IV.4 to bound the length of the sequence and thus the number of iterations by a function of k and ϵ, thereby completing the proof of Lemma V.2.

Initial Upper Bounds: We first show that the initial upper bounds we calculate in the algorithm are conservative in the sense that they are also respected by an optimal solution.

Lemma V.4. *If O is an optimal solution then $\delta(p, O) \le u(p)$ for any $p \in P$, where $u(p)$ is the initial upper bound computed in line 1 of Algorithm 1.*

Proof. Let $\alpha = u(p)$. For the sake of a contradiction, assume that $\delta(p, O) > \alpha$. By triangle inequality, any point $p' \in \mathsf{ball}(p, \alpha/3)$ has distance at least $2\alpha/3$ to O. Hence we have $\boldsymbol{\delta}(P, O) \ge (2\alpha/3) \cdot \mathbf{1}_{p,\alpha/3}$ and thus $f(\boldsymbol{\delta}(P, O)) \ge f((2\alpha/3) \cdot \mathbf{1}_{p,\alpha/3}) = (2\alpha/3) f(\mathbf{1}_{p,\alpha/3}) \ge (2\alpha/3) \cdot \frac{3\mathsf{OPT}}{\alpha} = 2\mathsf{OPT}$, which is a contradiction. □

The following lemma says that throughout the algorithm we approximately satisfy all upper bounds. We remark that the initialization (lines 2–7) as well as the analysis is a variant of Plesník's algorithm [23] for PRIORITY k-CENTER when applied to point set P with radii $u(p)$, $p \in P$.

Lemma V.5. *The number k' of points marked in line 3 in Algorithm 1 is at most k. Moreover, at any time during the execution of the while loop (lines 8–15), we have that $\delta(p, X) \le 4u(p)$. For any request (p, r) added to some cluster constraint, we have $r \le 4u(p)$.*

Proof. By Lemma V.4 each of the balls $\mathsf{ball}(p^{(\kappa)}, u(p^{(\kappa)}))$ with marked $p^{(\kappa)}$, $\kappa \in [k']$ contains at least one point from some hypothetical optimum solution O. On the other hand, these balls are pairwise disjoint by construction. Hence $k' \le |O| \le k$. This also implies that the algorithm can initialize $X = (x_1, \ldots, x_k)$ in line 7 with centers satisfying all initial cluster constraints. For example, it may pick the k' centers in F closest to $p^{(\kappa)}$, $\kappa \in [k']$ and $k - k'$ many additional arbitrary centers.

Because these initial requests are never removed, they are passed to the BALL INTERSECTION algorithm (with error parameter $\epsilon/10$; see line 14) whenever we make a change in the respective cluster. Hence, we have $\delta(p, X) \le (1 + \epsilon/10)u(p) \le 3u(p)/2$ for any marked point p throughout the execution of the while loop. For any point p' *not* marked, $\mathsf{ball}(p', u(p'))$ intersects $\mathsf{ball}(p, u(p))$ for some marked p. Because the points are processed in line 3 in non-decreasing order of $u(\cdot)$, we must have $u(p) \le u(p')$. As argued before, $\mathsf{ball}(p, 3u(p)/2)$ is guaranteed to contain a center in X at any time during the while loop. This center has distance at most $u(p') + 2 \cdot 3u(p)/2 \le 4u(p')$ from p' by triangle inequality. For the second claim, notice that $r < \delta(p, X) \le 4u(p)$ at the time this request is processed in line 14 for the first time. □

Bounding the Aspect-Ratio of Requests: The following lemma establishes that the radii of any two requests made for the same cluster are within a factor $\mathcal{O}(k/\epsilon^2)$ from each other. The intuition is as follows. We ensure in the algorithm (see line 10) that we only sample points whose radii are within a factor $\mathcal{O}(k/\epsilon)$ from $u(p)$. Assume that the radii, and thus the initial bounds $u(p)$, $u(p')$, in two request (p, r), (p', r') to the same cluster were very far from each other, say $r' \ll r$ and $u(p') \ll u(p)$. This would then imply that p was already (essentially) within radius r from some center before requesting (p, r) since there must be a center within radius $4u(p') \ll \epsilon r/3$ from p' by Lemma V.5. This contradicts the assumption that we requested (p, r) in the first place.

Lemma V.6. *Let (p, r) and (p', r') be requests added (in either order) to the same cluster constraint Q_κ, $\kappa \in [k]$ in line 13 of Algorithm 1. If $r' \le \epsilon^2 \cdot r/(10^4 k)$ then the algorithm fails in line 14 upon making the second of the two requests.*

Proof. Assume for the sake of a contradiction that the algorithm does not fail but finds a center x_κ with $\delta(p, x_\kappa) \le (1+\epsilon/10)r$ and $\delta(p', x_\kappa) \le (1+\epsilon/10)r'$. Hence $\delta(p, p') \le (1 + \epsilon/10)(r + r')$ by triangle inequality. By

Lemma V.5, we have $r \leq 4u(p)$ and $r' \leq 4u(p')$. Because we sample points from the set A defined in line 10, we have $r \geq \epsilon u(p)/(200k)$ and $r' \geq \epsilon u(p')/(200k)$.

Suppose $r' \leq \epsilon^2 r/(10^4 k)$. At the time of adding (p,r) to Q_κ the current candidate solution X satisfies $\delta(p', X) \leq 4u(p') \leq 1000kr'/\epsilon$ by Lemma V.5. Hence

$$\begin{aligned}\delta(p, X) &\leq \delta(p, p') + \delta(p', X) \\ &\leq (1 + {}^{\epsilon}\!/_{10})(r + r') + 1000kr'/\epsilon \\ &\leq (1 + {}^{\epsilon}\!/_{4})r\,.\end{aligned}$$

However, this is a contradiction because $\delta(p, X) = (1 + {}^{\epsilon}\!/_{3})r$ when requesting (p,r) to Q_κ as can be seen from line 14. □

Leveraging Bounded Algorithmic ϵ-Scatter Dimension: To complete the proof of Lemma V.2, we fix some cluster and consider the sequence of triples (x, p, r) where (p, r) is a request made for this cluster and where x is the center of the cluster just before the request was made. We establish that this sequence constitutes an algorithmic $(\mathcal{C}_\mathcal{M}, \epsilon)$-scattering and use Lemma V.6 to bound the aspect ratio of the radii in this sequence by $\mathcal{O}({}^{k}\!/_{\epsilon^2})$. We complete the proof via the aspect-ratio lemma IV.4.

Proof of Lemma V.2. Fix a cluster index $\kappa \in [k]$. Let $(p_\kappa^{(1)}, r_\kappa^{(1)}), \ldots, (p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ be the sequence of requests in the order in which they are added to Q_κ in line 13. For any $i \in [\ell]$, let $x_\kappa^{(i)}$ be the center of cluster κ at the time just before requesting $(p_\kappa^{(i)}, r_\kappa^{(i)})$. Since $r_\kappa^{(i)} = \delta(p_\kappa^{(i)}, X)/(1 + {}^{\epsilon}\!/_{3}) \leq \delta(p_\kappa^{(i)}, x_\kappa^{(i)})/(1 + {}^{\epsilon}\!/_{3})$ and since $x_\kappa^{(i)}$ is computed by invoking $\mathcal{C}$ on $\{(p_\kappa^{(1)}, r_\kappa^{(1)}), \ldots, (p_\kappa^{(i-1)}, r_\kappa^{(i-1)})\}$ and error parameter ${}^{\epsilon}\!/_{10}$, the sequence $(x_\kappa^{(1)}, p_\kappa^{(1)}, r_\kappa^{(1)}), \ldots, (x_\kappa^{(1)}, p_\kappa^{(\ell)}, r_\kappa^{(\ell)})$ is an algorithmic ${}^{\epsilon}\!/_{5}$-scattering.

By Lemma V.6, $r_\kappa^{(i)} \in R_\kappa = \left[r_{\min}, \frac{10^4 k r_{\min}}{\epsilon^2}\right]$ for every $i \in [\ell]$ where $r_{\min}$ denotes the smallest radius in any request for cluster κ. Applying Lemma IV.4 to the interval R_κ, the length of the sequence is $\mathcal{O}((\log {}^{k}\!/_{\epsilon})\lambda({}^{\epsilon}\!/_{10})/\epsilon)$. Since our algorithm adds in each iteration one request to some cluster constraint, the overall number of iterations is $\mathcal{O}(k(\log {}^{k}\!/_{\epsilon})\lambda({}^{\epsilon}\!/_{10})/\epsilon)$. □

2) Bounding the Success Probability: The proof of Lemma V.3 consists of two key steps: First, we argue that the algorithm terminates with success (that is, without failure) if the random choices made by the algorithm are "consistent" (to be defined more precisely below) with some hypothetical optimum solution. Second, we argue that consistency is maintained with sufficiently high probability in each iteration. Together with our upper bound on the number of iterations from Lemma V.2, this completes the proof of the main result, Theorem I.8.

Consistency: Informally speaking, we mean by consistency that a fixed hypothetical solution would satisfy all current cluster constraints.

Definition V.7. *Consider a fixed hypothetical optimum solution $O = (o_1, \ldots, o_k)$. We say that the current state of execution (specified by $(X, Q_1, \ldots, Q_k)$) of Algorithm 1 is* consistent with *O if for any request $(p, r) \in Q_\kappa$, $\kappa \in [k]$, we have that $\delta(p, o_\kappa) \leq r$.*

If the current state is consistent with the optimum solution O, then O certifies existence of solution to the cluster constraints $(Q_1, \ldots, Q_k)$ currently imposed. Therefore, the following observation is straightforward.

Observation V.8. *If the state of the algorithm is consistent with O before executing line 14 in any iteration, then the algorithm does not fail during this iteration.*

Probability of Maintaining Consistency: If the state of execution is consistent with O at the beginning of some iteration, then it remains consistent under the following two conditions. First, the point p sampled in this iteration is (randomly) assigned to the correct cluster. Second, the distance of p to the current candidate solution is sufficiently larger than its distance to O, thereby justifying the request made in line 13. This second condition motivates the following definition.

Definition V.9. *Given a candidate solution X with $f(\boldsymbol{\delta}(P, X)) > (1 + \epsilon)\mathit{OPT}$, a point $p \in P$ is called an* ϵ-witness *if $\delta(p, X) > (1 + \epsilon)\delta(p, O)$.*

The following lemma implies that the request made in any iteration for the sampled point is justified with probability $\Omega(\epsilon)$. It is a key part of our analysis as it links the specific way of (i) computing the initial upper bounds and (ii) sampling a witness based on these upper bounds. It is ultimately this interplay that allows us to bound the aspect ratio of the radii in the requests for a particular cluster and therefore the overall number of requests per cluster in terms of k and ϵ.

Lemma V.10. *Consider a fixed iteration of the while loop of Algorithm 1 and let X be the candidate solution at the beginning of this iteration. The point sampled in line 11 is then an ${}^{\epsilon}\!/_{3}$-witness for X with probability $\Omega(\epsilon)$. In particular, the set A computed in line 10 is not empty.*

Proof. For any subset $S \subseteq P$ of points let $C_S = \sum_{p \in S} w(p)\delta(p, X)$ denote the *contribution of S* towards $\boldsymbol{w}^\intercal \boldsymbol{\delta}(P, X) = C_P$.

Let $W \subseteq P$ be the subset of ${}^{\epsilon}\!/_{3}$-witnesses of X. We claim that the contribution C_W is at least $\epsilon C_P/10$. Suppose for the sake of a contradiction that their contribution is less. Then, using $0 < \epsilon < 1$,

$$\begin{aligned}\mathsf{OPT} &\geq \boldsymbol{w}^\intercal \boldsymbol{\delta}(P, O) \\ &\geq \sum_{p \in P \setminus W} w(p)\delta(p, O) \\ &\geq \frac{1}{1 + \epsilon/3} \sum_{p \in P \setminus W} w(p)\delta(p, X) \\ &\geq \frac{1 - \epsilon/10}{1 + \epsilon/3} \sum_{p \in P} w(p)\delta(p, X) \\ &\geq \frac{\boldsymbol{w}^\intercal \boldsymbol{\delta}(P, X)}{1 + \epsilon/2} \\ &\geq \frac{f(\boldsymbol{\delta}(P, X))}{(1 + \epsilon/2)(1 + \epsilon/10)} \\ &\geq \frac{f(\boldsymbol{\delta}(P, X))}{1 + 3\epsilon/4}\end{aligned}$$

which contradicts $f(\boldsymbol{\delta}(P, X)) > (1 + \epsilon)\mathsf{OPT}$.

Let $W_1, \ldots, W_k$ denote the subsets of the witnesses closest to centers $x_1, \ldots, x_k$ in X, respectively.

Let $H \subseteq [k]$ be the subset of clusters $\kappa \in [k]$ such that $C_{W_\kappa} \geq \epsilon C_P/(100k)$. Fix any cluster $\kappa \in H$. Let $\{z_1, \ldots, z_\ell\}$ be the witnesses in W_κ in non-decreasing order by the distance $\delta(z_i, x_\kappa)$, $i \in [\ell]$ to their closest cluster center x_κ. Let $j \in [\ell]$ be the minimum index j such that the contribution of the set $W_\kappa^- = \{z_1, \ldots, z_j\}$ is at least $C_{W_\kappa}/2$. This implies that also $C_{W_\kappa^+} \geq C_{W_\kappa}/2$ where $W_\kappa^+ = \{z_j, \ldots, z_\ell\}$. Hence $C_{W_\kappa^-}$ and $C_{W_\kappa^+}$ are both at least $\epsilon C_P/(200k)$ because $\kappa \in H$.

We claim that $W_\kappa^+ \subseteq A$ where A is defined as in line 10 in Algorithm 1. Towards this, let $p \in W_\kappa^+$ be arbitrary. We prove that $u(p) \leq 1000k\delta(p, x_\kappa)/\epsilon$ and hence $p \in A$. To see this, notice that $\mathsf{ball}(p, 2\delta(p, x_\kappa)) \supseteq \mathsf{ball}(x_\kappa, \delta(p, x_\kappa)) \supseteq W_\kappa^-$. On the other hand,

$$\begin{aligned}\frac{\epsilon\mathsf{OPT}}{300k} &\leq \frac{\epsilon f(\boldsymbol{\delta}(P, X))}{300k} \\ &\leq \frac{\epsilon C_P}{200k} \\ &\leq \sum_{q \in W_\kappa^-} w(q)\delta(q, x_\kappa) \\ &\leq \delta(p, x_\kappa) \sum_{q \in W_\kappa^-} w(q)\,.\end{aligned}$$

Setting $\alpha = 6\delta(p, x_\kappa)$, this implies that

$$\begin{aligned}f(\mathbf{1}_{p,\alpha/3}) &\geq \boldsymbol{w}^\intercal \mathbf{1}_{p,\alpha/3} \\ &= \sum_{q \in \mathsf{ball}(p,\alpha/3)} w(q) \\ &\geq \sum_{q \in W_\kappa^-} w(q) \\ &\geq \frac{\epsilon\mathsf{OPT}}{200k\delta(p, x_\kappa)} \\ &= \frac{\epsilon \cdot 3\mathsf{OPT}}{100k\alpha}\,.\end{aligned} \tag{1}$$

Hence $u(p) \leq 100k\alpha/\epsilon \leq 1000k\delta(p, x_\kappa)/\epsilon$ as claimed. This completes the proof of the claim that $W_\kappa^+ \subseteq A$ for any $\kappa \in H$.

As shown above, $\sum_{\kappa \in [k]} C_{W_\kappa} = C_W \geq \epsilon C_P/10$. By definition of H, we have $\sum_{\kappa \in [k] \setminus H} C_{W_\kappa} \leq \epsilon C_P/100$. Hence $\sum_{\kappa \in H} C_{W_\kappa} \geq \epsilon C_P/20$. Also, by the arguments above,

$$C_{A \cap W} \geq \sum_{\kappa \in H} C_{W_\kappa^+} \geq \sum_{\kappa \in H} \frac{C_{W_\kappa}}{2} \geq \frac{\epsilon C_P}{40} \geq \frac{\epsilon C_A}{40}\,.$$

Since we sample a point p from A with probability proportional to its contribution $C_{\{p\}}$, we sample a witness in each iteration with probability at least $\epsilon/40$.

Notice that $C_P \geq f(\boldsymbol{\delta}(P, X))/2 > 0$. The left-hand side of Equation 1 must therefore be positive. This implies that A is not empty. □

Overall Success Probability: We are now ready to prove Lemma V.3, thereby completing the proof of the main theorem I.8. We establish that the state of execution is consistent before entering the while loop in Algorithm 1. The proof is completed by combining the upper bound on the number of iterations (Lemma V.2) with the lower bound on the probability of maintaining consistence (Lemma V.10).

Proof of Lemma V.3. Let $p^{(1)}, \ldots, p^{(k')}$ be the points marked in line 3 of Algorithm 1. By Lemma V.4, each $\mathsf{ball}(p^{(\kappa)}, u(p^{(\kappa)}))$, $\kappa \in [k']$ contains a point from O. By construction, these balls are moreover pairwise disjoint. Hence, by relabeling the optimum centers $O = (o_1, \ldots, o_k)$, we can assume that $\delta(p^{(\kappa)}, o_\kappa) \leq u(p^{(\kappa)})$ for each marked point p where $\kappa \in [k']$ is the index of the cluster. Therefore the state of execution of the algorithm is consistent with O just before the first execution of the while loop (lines 8–15). Assume now that the state is consistent with O at the beginning of an iteration of the while loop. By Lemma V.10, we sample an $\epsilon/3$-witness p in this iteration with probability $\Omega(\epsilon)$. In this event, the request (p, r) added has radius $r = \delta(p, X)/(1 + \epsilon/3) \geq \delta(p, O)$. If additionally the cluster index $\kappa \in [k]$ picked at random is the same as the one in O—which happens with probability $\Omega(1/k)$—then the state remains consistent with O. In this event, the recomputation of the center in line 14 does not fail. By Lemma V.2, the algorithm terminates after at most $\mathcal{O}\left(\frac{k(\log k/\epsilon)\lambda(\epsilon/10)}{\epsilon}\right)$ many iterations. Since in any iteration it does not fail with probability $\Omega(\epsilon/k)$, it succeeds overall with probability $\exp\left(-\widetilde{\mathcal{O}}\left(\frac{k\lambda(\epsilon/10)}{\epsilon}\right)\right)$. □

VI. ϵ-Scatter Dimension Bounds

This section is devoted to bounding the ϵ-scatter dimension in various classes of metrics, proving Theorems I.3, I.4, and I.6 from the Introduction. Here, we outline the arguments and omit proofs, for which we refer to the full version [46].

A. Bounded Doubling Dimension

In this section, we show the upper bound of the ϵ-scatter dimension of any metric space of doubling dimension d, proving Theorem I.3.

a) Scatter Dimension and Packing: Given metric (X, δ), an ϵ-packing of this metric is a subset of points $X' \subseteq X$ such that $\delta(i,j) \geq \epsilon$ for all $i, j \in X'$. This is a standard notion in the theory of metric spaces. We first observe the following connection between our ϵ-scattering and ϵ-packing.

Observation VI.1. *Let $(x_1, p_1), \dots, (x_\ell, p_\ell)$ be an ϵ-scattering in a metric space (P, F, δ). Then, the set $X = \{x_2, \dots, x_\ell\}$ fo centers is an ϵ-packing in metric $(P \cup F, \delta)$ and X is contained in a unit ball.*

Corollary VI.2. *The size of ϵ-packing of a unit ball in metric M is at most the ϵ-scatter dimension minus one.*

It is a well-known fact that ϵ-packing of any metric of doubling dimension d has size at most $\mathcal{O}((1/\epsilon)^d)$. Combining this with Observation VI.1 yields Theorem I.3.

b) Remark: We note that the converse of Corollary VI.2 is false even in a very simple graph metric such as a star. In an n-node star rooted at r, a unit ball $\mathsf{ball}(r, 1)$ includes the whole graph. There exists an ϵ-packing of size $(n-1)$ by choosing the non-root nodes. However, any ϵ-scattering has length at most 2.

B. Bounded Treewidth Graphs

In this section we show that any graph of treewidth $\mathtt{tw}$ has ϵ-scatter dimension $\mathtt{tw}^{(1/\epsilon)^{\mathcal{O}(\mathtt{tw})}}$. That is, we prove Theorem I.4 for the bounded treewidth graph metric. We later show that the bound for planar graphs can be derived via an embedding result of [47]. For convenience, we abbreviate $\mathsf{ball}_{\delta_G}(r, \gamma)$ by $\mathsf{ball}_G(r, \gamma)$.

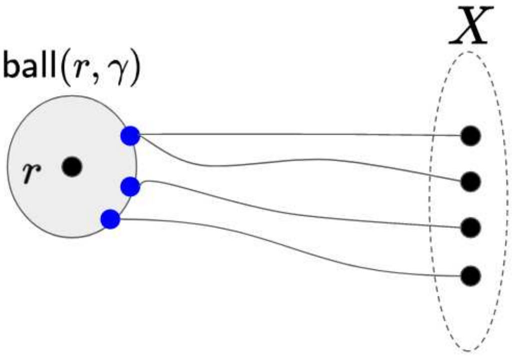

Fig. 4: A spider $S = \mathsf{ball}_G(r, \gamma)$ on X. Paths connecting X to r are disjoint, except for nodes in S.

1) Treewidth and Spiders: Our proof relies on the notion of spiders, whose existence can serve as a "witness" to the fact that the treewidth of a graph G is high. Given an edge-weighted graph G, $X \subseteq V(G)$ and $\gamma \in (0,1)$, a *γ-spider on X* is a set $S = \mathsf{ball}_G(r, \gamma)$ for some $r \in V(G)$ such that there are $|X|$ paths from S to X that are vertex-disjoint except for in S. We say that a set S is a *spider* on X if it is a γ-spider for some γ. See Figure 4 for illustration.

Observe that if S is a γ-spider on X, then for any $X' \subseteq X$, S is also a γ-spider on X'. The following lemma is key to our result, roughly showing that the existence of a large number of spiders implies that the treewidth of G is large.

Lemma VI.3. *Let G be a graph, k be an integer and $X \subseteq V(G) : |X| > 3k$. If there is a family $\mathcal{S}$ of $k+1$ pairwise disjoint spiders on X, then the treewidth of G is larger than k.*

2) Iteratively Finding Spiders: Our main result in this section is encapsulated in the following theorem.

Theorem VI.4. *If there is an ϵ-scattering of length at least $(\mathcal{O}(k/\epsilon))^{(4/\epsilon)^{k+1}}$ in G, then graph G contains a family of $k+1$ disjoint spiders on vertex set of size greater than $3k$.*

Combining the above with Lemma VI.3, we can deduce that the length of any ϵ-scattering is at most $\mathtt{tw}^{(1/\epsilon)^{\mathcal{O}(\mathtt{tw})}}$ as desired. We spend the rest of this section proving the theorem. Given ϵ-scattering σ, we say that the ϵ-packing $X = X(\sigma)$, given by Observation VI.1, is a canonical packing of σ.

Lemma VI.5. *Let σ be an ϵ-scattering of length ℓ in $G \subseteq \mathsf{ball}_G(r, 1)$ and $X = X(\sigma)$ its canonical ϵ-packing. Then, there exist*

- *a spider $S = \mathsf{ball}_G(r, \epsilon/3)$ on $X' \subseteq X : |X'| \geq$*

$c_0\epsilon \cdot (\ell/2)^{\epsilon/3}$ for some constant c_0 and

- *a graph G' such that $S \cap V(G') = \emptyset$ and an ϵ-scattering σ' that is a subsequence of σ such that $X(\sigma') = X'$.*

We show how this lemma implies Theorem VI.4. Let $G_0 = G$ contain a ϵ-scattering σ_0 of length at least $\ell_0 = (\frac{k}{c_0\epsilon})^{(4/\epsilon)^{k+1}}$ and $X_0 = X(\sigma_0)$. The lemma allows us to find a spider S_1 on X_1 of size $c_0\epsilon \cdot \ell_0^{\epsilon/3} \geq (\frac{k}{c_0\epsilon})^{(4/\epsilon)^{k}} = \ell_1$ for sufficiently small ϵ. Moreover, we have the graph G_1 that is disjoint with S_1 and ϵ-scattering that is a subsequence σ_1 of length ℓ_1. Since (G_1, X_1, σ_1) satisfies the preconditions of Lemma VI.5, we can apply it to obtain (G_2, X_2, σ_2) and so on. More formally, starting from (G_i, σ_i, X_i), we apply Lemma VI.5 to obtain $(G_{i+1}, \sigma_{i+1}, X_{i+1})$. We maintain the following invariant: The length of the sequence σ_i satisfies $\ell_i = |X_i| \geq (\frac{k}{c_0\epsilon})^{(4/\epsilon)^{k+1-i}}$. This allows us to find disjoint spiders $S_1, S_2, \ldots, S_{k+1}$ on $X_{k+1} : |X_{k+1}| > 3k$ as desired.

C. Bounding ϵ-Scatter Dimension via Low-Treewidth Embedding

In this section, we show a (simple) connection between bounding ϵ-scatter dimension and an active research area on embedding with additive distortion [16], [40], [41]. This connection allows us to upper bound the ϵ-scatter dimension of planar graphs.

In particular, we say that (weighted) graph class $\mathcal{G}$ admits a *t-low treewidth-diameter embedding* for function $t \colon \mathbb{N} \to \mathbb{N}$ if there exists a *deterministic* algorithm that takes G and produces a weighted graph H of treewidth at most $t(\eta)$ and an embedding $\phi \colon V(G) \to V(H)$ such that:

$$\delta_G(u, v) \leq \delta_H(\phi(u), \phi(v)) \leq \delta_G(u, v) + \eta D$$

where D is the diameter of G.

Theorem VI.6. *Let $\lambda_{\mathtt{tw}}(\epsilon)$ denote the the ϵ-scatter dimension of graphs of treewidth $\mathtt{tw}$ (from the previous section, this bound is at most doubly exponential in $\mathtt{tw}$). If graph class $\mathcal{G}$ admits a t-low treewidth-diameter embedding, then every metric in $\mathcal{G}$ has ϵ-scatter dimension at most $\lambda_{t(\epsilon/10)}(\epsilon/3)$.*

Now we can use the following theorem.

Theorem VI.7 (Theorem 1.3 of [16])**.** *There is a polynomial-time algorithm that, given an edge-weighted planar graph and given a number $\eta > 0$, outputs an embedding of the graph into a planar graph of treewidth $\mathsf{poly}(1/\eta)$ with additive error $\eta \cdot D$ where D is the diameter of the input graph.*

This implies, in our language, that planar graphs have low treewidth-diameter embedding.

Corollary VI.8. *Planar graphs have ϵ-scatter dimension at most* $\exp\left(\exp(\mathsf{poly}(1/\epsilon))\right)$.

D. High-Dimensional Euclidean Space

Recall, from the introduction and Sections IV, that the ϵ-scatter dimension of high-dimensional (continuous) Euclidean space is unbounded. In this section, we show that, in contrast, the algorithmic ϵ-scatter dimension of this metric is bounded.

Theorem I.7 (Bounding Algorithmic Scatter Dimension)**.** *The continuous Euclidean space (P, F, δ), that is, $P \subsetneq \mathbb{R}^d$ finite, and $F = \mathbb{R}^d$, has algorithmic ϵ-scatter dimension $\mathcal{O}(1/\epsilon^4 \log 1/\epsilon)$.*

We outline the proof of Theorem I.7. In order to upper bound the algorithmic ϵ-scatter dimension for the continuous Euclidean space, it suffices to show that there exists an algorithm $\mathcal{C}$ such that the $(\mathcal{C}, \epsilon)$-scattering dimension in the Euclidean space is bounded. We use an algorithm by Kumar and Yildirim [43] as BALL INTERSECTION algorithm for the high-dimensional Euclidean space. They study the BALL INTERSECTION problem in the language of WEIGHTED EUCLIDEAN 1-CENTER. They provide a BALL INTERSECTION algorithm based on a convex optimization formulation which efficiently (and approximately) solves the BALL INTERSECTION problem in continuous Euclidean setting for weights with bounded aspect ratio. Let $\mathcal{C}_{\text{KY}}$ denote this algorithm. The following lemma is adapted from Kumar and Yildirim's work into our terminology (see Lemma 4.2 of [43]).

Lemma VI.9. *Given an instance (P, F, δ) of* BALL INTERSECTION *in high-dimensional Euclidean space, associated radii $r(p)$ to each $p \in P$, and $\epsilon \in (0, 1)$, the length of any $(\mathcal{C}_{\text{KY}}, \epsilon)$-scattering is at most $\mathcal{O}\left(\tau/\epsilon^2\right)$ where $\tau \geq 1$ is the squared ratio of the largest radius in the requests to the smallest.*

Note that for a constant τ, Lemma VI.9 already gives the theorem. To complete the argument for Theorem I.7 in the general setting, we show that by increasing the length of the ϵ-scattering by a multiplicative factor of $\mathcal{O}\left(\log 1/\epsilon\right)$, we can assume that τ is $\mathcal{O}\left(1/\epsilon^2\right)$.

Aspect-Ratio Condition: The following lemma provides a sufficient condition for bounded algorithmic ϵ-scatter dimension that facilitates the design of a BALL INTERSECTION algorithm for bounding the algorithmic ϵ-scatter dimension. In particular, this condition is key to bound the algorithmic ϵ-scatter dimension of high-dimensional continuous Euclidean spaces. It can be seen as a strenghtened converse of the aspect-ratio lemma V.6 and holds for arbitrary classes of metric spaces.

Lemma VI.10 (Aspect-Ratio Condition)**.** *Let $\mathcal{M}$ be a class of metric spaces with* BALL INTERSECTION *algorithm $\mathcal{C}_{\mathcal{M}}$ and let $\epsilon \in (0,1)$. If any $(\mathcal{C}_{\mathcal{M}}, \epsilon)$-scattering $(x_1, p_1, r_1), \ldots, (x_\ell, p_\ell, r_\ell)$ with $r_i \in [\epsilon/12, 1]$, $i \in [\ell]$ contains at most $\lambda(\epsilon)$ triples with the same radius, then the algorithmic ϵ-scatter dimension of $\mathcal{M}$ bounded by $\mathcal{O}(\lambda(\epsilon) \log 1/\epsilon)$.*

Lemmas VI.9 and VI.10 together with the observation that $\tau = (12/\epsilon)^2$ give the proof of Theorem I.7.

VII. Conclusions and Open Problems

We present a unified view on efficient parameterized approximation schemes that applies to large variety of clustering objectives and metric spaces. From complexity theoretic perspective, this implies rather surprising collapses of approximability of symmetric and asymmetric norm clustering problems (in the regime of **P**, their approximabilities are substantially different, and yet they collapse in **FPT**).

There are rooms for future open problems in two directions that can be pursued independently. First, can we characterize the class of metric spaces with bounded scatter dimension? For example, do minor-free graphs have bounded scatter dimension? This is a purely structural question whose resolution immediately yields an EPAS (through our framework). Since the bounded treewidth graphs play an important role in our approach (through the lens of low treewidth embedding), it would be interesting to pinpoint the exact bound on their ϵ-scatter dimension; a particularly interesting concrete question is whether the bound can be brought down to singly exponential.

The second direction concerns extensions of our framework. Some clustering objectives are still missing from our framework. For instance, what about clustering with outliers [48], [49], [50] (in which case the cost function f is instead an anti-norm)? Even more conceptually, our current algorithm is oblivious to the structure of the input metric, but our theorem can only talk about whether an EPAS can be obtained. Is it possible for such a framework to give approximation factors in all spectrums (e.g., $(3 + o(1))$-approximation for k-Center if a general, unstructured metric space is given as input)? The "dream result" could look something like $(\gamma(\mathcal{M}) + o(1))$ FPT approximation algorithm where $\gamma(\mathcal{M})$ is the FPT-approximability of metric class $\mathcal{M}$.

Last, but not least, the scope of this paper is to handle multiple clustering objectives and metric spaces. Many clustering problems additionally impose restrictions on how points in P can be assigned to open centers in X, e.g., capacity [51], [52], [53], different notions of fairness [54], [55], [56] and diversity constraints [57], [58], [59]; in such case, our framework does not apply. Extending our framework to handle such constraints (or proving that EPASes do not exist when such constraints are enforced) is an interesting direction.

References

[1] A. Kumar, Y. Sabharwal, and S. Sen, "Linear-time approximation schemes for clustering problems in any dimensions," *J.ACM*, vol. 57, no. 2, pp. 1–32, 2010.

[2] D. Feldman, M. Monemizadeh, and C. Sohler, "A PTAS for k-means clustering based on weak coresets," in *Proceedings of the twenty-third annual symposium on Computational geometry (SoCG'07)*, 2007, pp. 11–18.

[3] S. Dasgupta, "The hardness of k-means clustering," University of California, San Diego, San Diego, CA, Technical Report CS2008-0916, 2008.

[4] P. Awasthi, M. Charikar, R. Krishnaswamy, and A. K. Sinop, "The hardness of approximation of euclidean k-means," in *31st International Symposium on Computational Geometry (SoCG'15)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[5] S. Har-Peled and S. Mazumdar, "On coresets for k-means and k-median clustering," in *Proceedings of the thirty-sixth annual ACM Symposium on Theory of computing (STOC'04)*, 2004, pp. 291–300.

[6] J. Matoušek, "On approximate geometric k-clustering," *Discrete & Computational Geometry*, vol. 24, no. 1, pp. 61–84, 2000.

[7] R. Ostrovsky, Y. Rabani, L. J. Schulman, and C. Swamy, "The effectiveness of Lloyd-type methods for the k-means problem," *J.ACM*, vol. 59, no. 6, pp. 1–22, 2013.

[8] H. Ding and J. Xu, "A unified framework for clustering constrained data without locality property," *Algorithmica*, vol. 82, no. 4, pp. 808–852, 2020.

[9] A. Bhattacharya, R. Jaiswal, and A. Kumar, "Faster algorithms for the constrained k-means problem," *Theory of Computing Systems*, vol. 62, no. 1, pp. 93–115, 2018.

[10] V. Braverman, S. H.-C. Jiang, R. Krauthgamer, and X. Wu, "Coresets for clustering in excluded-minor graphs and beyond," in *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA'21)*. SIAM, 2021, pp. 2679–2696.

[11] V. Cohen-Addad, M. Pilipczuk, and M. Pilipczuk, "Efficient approximation schemes for uniform-cost clustering problems in planar graphs," in *27th Annual European Symposium on Algorithms (ESA'19)*, vol. 144. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019, pp. 33–1.

[12] V. Cohen-Addad and C. Karthik, "Inapproximability of clustering in lp metrics," in *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS'19)*. IEEE, 2019, pp. 519–539.

[13] M. Badŏiu, S. Har-Peled, and P. Indyk, "Approximate clustering via core-sets," in *Proc. 34th Annual ACM Symposium on Theory of Computing (STOC'02)*, J. H. Reif, Ed. ACM, 2002, pp. 250–257. [Online]. Available: https://doi.org/10.1145/509907.509947

[14] D. Baker, V. Braverman, L. Huang, S. H.-C. Jiang, R. Krauthgamer, and X. Wu, "Coresets for clustering in graphs of bounded treewidth," in *International Conference on Machine Learning (ICML'20)*. PMLR, 2020, pp. 569–579.

[15] I. Katsikarelis, M. Lampis, and V. T. Paschos, "Structural parameters, tight bounds, and approximation for (k, r)-center," *Discrete Applied Mathematics*, vol. 264, pp. 90–117, 2019.

[16] E. Fox-Epstein, P. N. Klein, and A. Schild, "Embedding planar graphs into low-treewidth graphs with applications to efficient approximation schemes for metric problems," in *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'19)*. SIAM, 2019, pp. 1069–1088.

[17] D. Chakrabarty and C. Swamy, "Approximation algorithms for minimum norm and ordered optimization problems," in *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC'19)*, 2019, pp. 126–137.

[18] L. Huang and N. K. Vishnoi, "Coresets for clustering in euclidean spaces: importance sampling is nearly optimal," in *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC'20)*, 2020, pp. 1416–1429.

[19] V. Cohen-Addad, D. Saulpic, and C. Schwiegelshohn, "A new coreset framework for clustering," in *Proceedings of the 53rd*

Annual ACM SIGACT Symposium on Theory of Computing (STOC'21), 2021, pp. 169–182.

[20] V. Cohen-Addad, K. G. Larsen, D. Saulpic, and C. Schwiegelshohn, "Towards optimal lower bounds for k-median and k-means coresets," in *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC'22)*, 2022, pp. 1038–1051.

[21] I. Li Gørtz and A. Wirth, "Asymmetry in k-center variants," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX'03)*. Springer, 2003, pp. 59–70.

[22] T. Bajpai, D. Chakrabarty, C. Chekuri, and M. Negahbani, "Revisiting priority k-center: Fairness and outliers," in *48th International Colloquium on Automata, Languages, and Programming (ICALP'21)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021, p. 21.

[23] J. Plesník, "A heuristic for the p-center problems in graphs," *Discrete Applied Mathematics*, vol. 17, no. 3, pp. 263–268, 1987.

[24] V. Cohen-Addad, A. Gupta, A. Kumar, E. Lee, and J. Li, "Tight FPT approximations for k-median and k-means," in *Proc. 46th International Colloquium on Automata, Languages, and Programming (ICALP'19)*, ser. LIPIcs, vol. 132, 2019, pp. 42:1–42:14. [Online]. Available: https://doi.org/10.4230/LIPIcs.ICALP.2019.42

[25] A. Tamir, "The k-centrum multi-facility location problem," *Discrete Applied Mathematics*, vol. 109, no. 3, pp. 293–307, 2001.

[26] S. Nickel and J. Puerto, *Location Theory*. Springer Science & Business Media, 2005.

[27] G. Laporte, S. Nickel, and F. S. da Gama, *Location Science*. Springer, 2015.

[28] J. Byrka, K. Sornat, and J. Spoerhase, "Constant-factor approximation for ordered k median," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, 2018, pp. 620–631.

[29] V. Braverman, S. H.-C. Jiang, R. Krauthgamer, and X. Wu, "Coresets for ordered weighted clustering," in *International Conference on Machine Learning (ICML'19)*. PMLR, 2019, pp. 744–753.

[30] M. Abbasi, A. Bhaskara, and S. Venkatasubramanian, "Fair clustering via equitable group representations," in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT'21)*, 2021, pp. 504–514.

[31] D. Goyal and R. Jaiswal, "Tight fpt approximation for socially fair clustering," *Information Processing Letters*, p. 106383, 2023.

[32] Y. Makarychev and A. Vakilian, "Approximation algorithms for socially fair clustering," in *Conference on Learning Theory (COLT'21)*. PMLR, 2021, pp. 3246–3264.

[33] M. Ghadiri, M. Singh, and S. S. Vempala, "Constant-factor approximation algorithms for socially fair k-clustering," *arXiv preprint arXiv:2206.11210*, 2022.

[34] B. Anthony, V. Goyal, A. Gupta, and V. Nagarajan, "A plant location guide for the unsure: Approximation algorithms for min-max location problems," *Mathematics of Operations Research*, vol. 35, no. 1, pp. pages 79–101, 2010.

[35] S. Bhattacharya, P. Chalermsook, K. Mehlhorn, and A. Neumann, "New approximability results for the robust k-median problem," in *Scandinavian Workshop on Algorithm Theory (SWAT'14)*. Springer, 2014, pp. 50–61.

[36] N. Bansal, R. Khandekar, J. Könemann, V. Nagarajan, and B. Peis, "On generalizations of network design problems with degree bounds," *Mathematical Programming*, vol. 141, no. 1, pp. 479–506, 2013.

[37] E. Chlamtáč, Y. Makarychev, and A. Vakilian, "Approximating fair clustering with cascaded norm objectives," in *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'22)*. SIAM, 2022, pp. 2664–2683.

[38] F. Abbasi, S. Banerjee, J. Byrka, P. Chalermsook, A. Gadekar, K. Khodamoradi, D. Marx, R. Sharma, and J. Spoerhase, "Parameterized approximation for robust clustering in discrete geometric spaces," *CoRR*, vol. abs/2305.07316, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2305.07316

[39] K. L. Clarkson, "Nearest-neighbor searching and metric space dimensions," *Nearest-neighbor methods for learning and vision: theory and practice*, pp. 15–59, 2006.

[40] A. Filtser and H. Le, "Low treewidth embeddings of planar and minor-free metrics," in *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS'22)*. IEEE, 2022, pp. 1081–1092.

[41] V. Cohen-Addad, A. Filtser, P. N. Klein, and H. Le, "On light spanners, low-treewidth embeddings and efficient traversing in minor-free graphs," in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS'20)*. IEEE, 2020, pp. 589–600.

[42] M. Grötschel, L. Lovász, and A. Schrijver, *Geometric algorithms and combinatorial optimization*. Springer Science & Business Media, 2012, vol. 2.

[43] P. Kumar and E. A. Yildirim, "An algorithm and a core set result for the weighted euclidean one-center problem," *INFORMS Journal on Computing*, vol. 21, no. 4, pp. 614–629, 2009. [Online]. Available: https://doi.org/10.1287/ijoc.1080.0315

[44] L. Lovász and S. Vempala, "Fast algorithms for logconcave functions: Sampling, rounding, integration and optimization," in *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*. IEEE, 2006, pp. 57–68.

[45] Y. T. Lee, A. Sidford, and S. S. Vempala, "Efficient convex optimization with membership oracles," in *Conference On Learning Theory (COLT'18)*. PMLR, 2018, pp. 1292–1294.

[46] F. Abbasi, S. Banerjee, J. Byrka, P. Chalermsook, A. Gadekar, K. Khodamoradi, D. Marx, R. Sharma, and J. Spoerhase, "Parameterized approximation schemes for clustering with general norm objectives," *CoRR*, vol. abs/2304.03146, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2304.03146

[47] Z. Friggstad, M. Rezapour, and M. R. Salavatipour, "Local search yields a PTAS for k-means in doubling metrics," *SIAM J.Computing*, vol. 48, no. 2, pp. 452–480, 2019.

[48] R. Krishnaswamy, S. Li, and S. Sandeep, "Constant approximation for k-median and k-means with outliers via iterative rounding," in *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing (STOC'18)*, I. Diakonikolas, D. Kempe, and M. Henzinger, Eds. ACM, 2018, pp. 646–659.

[49] S. Banerjee, R. Ostrovsky, and Y. Rabani, "Min-sum clustering (with outliers)," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, (APPROX/RANDOM'21)*, ser. LIPIcs, vol. 207. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021, pp. 16:1–16:16.

[50] S. Deng and Q. Zhang, "Ordered k-median with outliers," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, (APPROX/RANDOM'22)*, ser. LIPIcs, A. Chakrabarti and C. Swamy, Eds., vol. 245. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, pp. 34:1–34:22.

[51] M. Adamczyk, J. Byrka, J. Marcinkowski, S. M. Meesum, and M. Włodarczyk, "Constant-factor FPT approximation for capacitated k-median," in *27th Annual European Symposium on Algorithms (ESA'19)*. Schloss Dagstuhl-Leibniz-Zentrum fur Informatik GmbH, Dagstuhl Publishing, 2019, p. 1.

[52] H. G. Demirci and S. Li, "Constant approximation for capacitated k-median with $(1+\epsilon)$-capacity violation," in *43rd International Colloquium on Automata, Languages, and Programming (ICALP'16)*, ser. LIPIcs, vol. 55, 2016, pp. 73:1–73:14.

[53] V. Cohen-Addad, "Approximation schemes for capacitated clustering in doubling metrics," in *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms (SODA'20)*, S. Chawla, Ed. SIAM, 2020, pp. 2241–2259.

[54] S. K. Bera, D. Chakrabarty, N. Flores, and M. Negahbani, "Fair algorithms for clustering," in *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems (NeurIPS'19)*, 2019, pp. 4955–4966.

[55] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (NIPS'17)*, 2017, pp. 5029–5037.

[56] X. Chen, B. Fain, L. Lyu, and K. Munagala, "Proportionally fair clustering," in *Proceedings of the 36th International Conference on Machine Learning (ICML'19)*, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 2019, pp. 1032–1041.

[57] J. Li, K. Yi, and Q. Zhang, "Clustering with diversity," in *Automata, Languages and Programming, 37th International Colloquium (ICALP'19)*, vol. 6198. Springer, 2010, pp. 188–200.

[58] S. Thejaswi, B. Ordozgoiti, and A. Gionis, "Diversity-aware k-median: Clustering with fair center representation," in *Machine Learning and Knowledge Discovery in Databases. Research Track - European Conference (ECML PKDD'21), Bilbao, Spain, September 13-17, 2021, Proceedings, Part II*, vol. 12976. Springer, 2021, pp. 765–780.

[59] S. Thejaswi, A. Gadekar, B. Ordozgoiti, and M. Osadnik, "Clustering with fair-center representation: Parameterized approximation algorithms and heuristics," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '22)*. New York, NY, USA: Association for Computing Machinery, 2022, p. 1749–1759. [Online]. Available: https://doi.org/10.1145/3534678.3539487